Mitigating Denial-of-Service Attacks in MANET by Incentive-based Packet Filtering: A Game-theoretic Approach

by Xiaoxin Wu, David K. Y. Yau

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

Mitigating Denial-of-Service Attacks in MANET by Incentive-based Packet Filtering: A Game-theoretic Approach

Xiaoxin Wu
Intel Communication Beijing Lab
Beijing, China
Email: xiaoxin.wu@intel.com

David K. Y. Yau
Department of Computer Science
Purdue University
West Lafayette, IN 47907, USA
Email: yau@cs.purdue.edu

***Abstract*—Defending against denial-of-service attacks (DoS) in a mobile ad hoc network (MANET) is challenging because the network topology is dynamic and nodes are selfish. In this paper, we propose a DoS mitigation technique that uses digital signatures to verify legitimate packets, and drop packets that do not pass the verification. Since nodes are selfish, they may not perform the verification in order to avoid paying the overhead. A bad packet that escapes verification along the whole network path will bring a penalty to all its forwarders. A network game can be formulated in which nodes along a network path, in optimizing their own benefits, are encouraged to act collectively to filter out bad packets. Analytical results show that Nash equilibrium can be attained for players in the proposed game, and significant benefits can be provided to forwarders such that many of the bad packets will be eliminated by verification.**

I. Introduction

The dependencies between dynamic, mutually untrusted neighbors in a mobile ad hoc network (MANET) create important security concerns in these networks. Among the attacks documented in the literature, denial-of-service (DoS) attacks are particularly damaging since both communication bandwidth and node resources are scarce in MANETs. In addition to their ability to take down a network quickly, DoS attacks directed at bandwidth and end node resources are easy to launch; e.g., by simply injecting useless traffic into the network.

DoS mitigation techniques designed for wireline networks, such as traceback by stamping [15], [16] or filtering [8], [2] packets, and capability based network access [20], [19], will not work well in an ad hoc environment where the routes and the set of forwarders on a routing path are highly dynamic and are selfish.

Secure routing protocols designed for ad hoc networks [7] build secure routes to support end-to-end communication. If link layer security is applied [10], these protocols can mitigate DoS attacks. Illegitimate packets will be discovered as outside attackers do not know the keys shared between the hops. However, an inside attacker, i.e., an attacker who is a member of the end-to-end path, can still launch an attack. Without using signatures, it is difficult to identify the attacker even if the attacker is known to be an insider. In addition, in networks where packet delivery is route-based (e.g., GPSR [9]), these secure routing protocols cannot be applied because the path can change from packet to packet.

Motivating nodes to serve each other is another fundamental issue in MANETs. Specifically, as communication endpoints rely on intermediate nodes to forward their traffic, incentives for the forwarders have to be provided. Traditional incentive systems have used nuggets [4] and reputation credits [3] to encourage nodes to function as forwarders. The incentive issue becomes even more relevant in the security context, when security measures may require certain nodes to expend more resources to better defend other nodes. The incentive issue as it relates to the security issue has been less addressed by the research community.

Game theory has been applied in MANET for solving the incentive problem in multi-hop forwarding [5] [21] [6] [17]. In [1], a game model of network attacks is developed for the interactions between an intrusion detection system (IDS) and an attacker with dynamic system information. Applying game theory to induce collaboration between network nodes for improved security has been less addressed in the literature.

In this work, we propose a DoS mitigating technique for MANETs that jointly considers the security and incentive issues. The technique is designed to work in a packet-switching network environment. The idea is based on an attacker's goal to avoid detection and being identified. Hence, we protect legitimate packets by requiring them to be signed by their respective senders. A forwarder verifies a packet's sender signature when the packet is received. If the verification fails, the packet is dropped. Otherwise, it is forwarded.

We assume that network nodes are selfish but rational. Incentive for a node to forward packets is given by a reward the node will obtain after the packets are successfully delivered to their final destinations. A forwarder may also choose to forward a packet without verification, since the operation carries a cost. To motivate a forwarder to verify, a penalty is assessed for a "lazy" node each time it forwards an attacker packet that finally reaches the destination. We will investigate the properties of the resulting game, as forwarders independently attempt to play a best forwarding/verification strategy that will maximize their own payoffs, while the network is subject to given inputs of attacking and legitimate traffic.

Our main contributions in this paper are as follows:

- We propose a signature-based DoS mitigation technique for packet-switching MANETs.
- We use game theory to study how a system of forwarders can be motivated to forward good packets while filtering out bad packets cooperatively by verification. We will propose solutions that address jointly the security and incentive issues. We will discuss how practical cost functions can be assigned for sending, forwarding, receiving, and verifying packets. In addition, we address implementation issues for an accounting system that will support the proposed system.
- We present game-theoretic analysis results to investigate the impact of system parameters on the game's performance and the resulting behaviors of individual nodes. The results serve as useful guidelines for network management.

The balance of the paper is organized as follows. In In Section II, we introduce the use of distributed filtering for DoS mitigation and discuss its game rationality. In Section III, we present the detailed formulation of our game-theoretic DoS mitigation system. In Section IV, we address implementation issues for the proposed system. Simulation results are presented in Section V to illustrate the system's performance. Section VI concludes.

II. Game Theoretic DOS Mitigation in MANET

A. Assumptions and attack model

Network Assumptions. We assume that the network consists of a number of ad hoc nodes. The network is loosely synchronized, e.g, using the method in [14]. Nodes are selfish but rational. A node can be a receiver (i.e., destination), a sender (i.e., source), or an intermediate forwarder. Data packets are delivered from a source towards their destinations in a packet-switching manner. Connections are multi-hop. Sessions are short, so no virtual route has to be built for data delivery. Upon receiving a packet, a forwarder determines its next hop based on the destination address, and forwards the packet to the next hop. Determining the next hop is a function of the routing algorithm, such as position-based routing. As nodes are mobile, the path between a given source/destination pair of nodes may change.

Security Assumptions. We assume that a PKI local to the deployment network is in place when the network is set up. For example, a trusted party, such as a Certification Authority (CA), possesses a pair of public/private keys, where the public key is configured with each ad hoc node. Each ad hoc node also possesses a pair of public/private keys. The public key is certified by the CA, and is paired with the node's identity. Although there may exist an end-to-end symmetric key between a source and a destination, we do not assume link layer symmetric keys between any two neighboring nodes; i.e., the link level information is transmitted in plain text. Lastly, we assume that the network is aware of the severity of the DoS attack; i.e., an approximate fraction of the total packets that are attack packets.

Payment Model. We assume that there is an off-line accounting system that handles the payment/penalty with respect to data delivery. The payment is well defined for each transaction conducted by ad hoc users. We assume that the accounting system is strong enough to distinguish a legitimate ad hoc user from a cheater who claims credits that it does not deserve. Implementation issues for the accounting system are discussed in Section IV.

Attack Model. We assume that a number of attackers, which are otherwise ordinary ad hoc nodes[1], may inject traffic (i.e., bad packets) in the network to trigger a DoS attack. The impact of the attack is two-fold: (1) it consumes excessive receiver resources by, say, creating a large number of computing processes at the receiver; and (2) it consumes excessive network bandwidth to congest the target network. In this paper, we focus on the case when an attacker randomly selects victims. Unless otherwise specified, we assume that the attackers do not collude. Other attacks such as packet dropping and packet manipulation are orthogonal to the jamming attack, and are not covered in this work.

B. Mitigating DoS in MANET

We require that packets from legitimate sources be digitally signed by their respective senders. Other than the network level routing information and the application level data payload, each packet will carry a signature (SIG), which includes the hash result of the packet signed by the private key of its source and a certificate for the corresponding public key. The signed SIG with the certificate is used to verify that the packet is from the claimed legitimate source. If the SIG carried in the packet does not match the SIG that a forwarder generates from the received packet, the packet is classified as a bad packet and therefore dropped.

1) Against Replay Attack: The signature-based defense is prone to the replay attack. An attacker can replay a legitimate packet a large number of times to generate a high load of useless traffic. These packets will pass the verification step. To deal with the replay attack, a packet should be stamped with its generation time. In addition, each packet has a given lifetime. A packet whose lifetime has expired will be dropped. To prevent a malicious node from sending a legitimate packet to different next hops during the packet's lifetime, a *neighbor monitoring* technique can be used.

In neighbor monitoring, a node reads the complete header, including both the SIG and network level headers, of every packet even if the node is not the packet's next hop. The node stores the header read until the corresponding packet's lifetime expires. Upon hearing a packet whose lifetime has not expired, the node will compare the header read with the headers currently in the node's local store. By doing this, the node can detect a replayed packet and drop it before further

[1] That is, the attackers are not superior to ordinary ad hoc nodes.

damage to the network happens. Since only the packet header, but not the whole packet, has to be read, the cost of monitoring can be kept low. If the packet lifetime is not too long, which is normally the case in ad hoc networks, a node will not need to store too many packet headers, which reduces the storage cost. Note that the monitoring technique will not be effective in a wireline network if attackers select different routes for sending different replayed packets, since one forwarder will then be unable to monitor packets destined for another forwarder.

As shown in Fig. 1 in a two-dimensional plane, we illustrate how neighbor monitoring mitigates non-collusive replay attacks in a network with high node density. If position-based routing is used for packet delivery, as the node density is high, the path for any packet delivery can be approximated as a straight line, and the geographical distance for each hop is approximately the same as the maximum range of radio transmission, denoted as R. For example, when a forwarder F at $(0, 0)$ wishes to send a packet to the destination D, it may do so through a next hop N, where $FN \approx R$. When neighbor monitoring is applied, any node that is no more than distance R from the path (i.e., $... \rightarrow F \rightarrow N... \rightarrow D$) will be aware of the packet's forwarding to N. In Fig. 1, any node located in the shaded rectangular area knows about the forwarding.

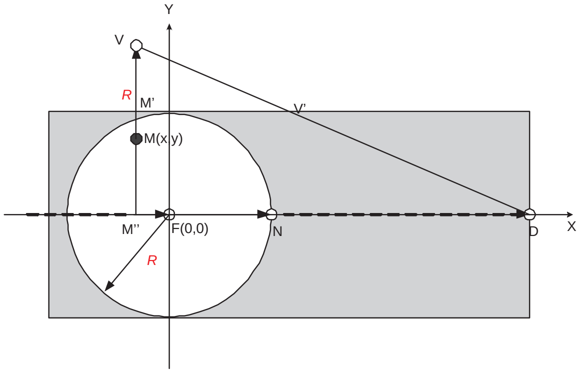


Fig. 1. Mitigating replay attack using neighbor monitoring.

We assume that a malicious attacker at $M(x, y)$ intercepts the packet from the forwarder located at F. To avoid detection of the replayed packet by any node located in the shaded area in Fig. 1, the attacker would find a node that is the farthest from the packet's delivery path as the next hop. We assume that this next hop is located at V. Consequently, $MV \approx R$. Note that MV is perpendicular to FD. As V is well behaving, it will forward the packet towards D following the path FD. The replayed packet will be discovered once it crosses the point V' (i.e., on entering the shaded area in Fig. 1). Hence, how successful the replay attack is depends on the ratio between VV' and VD, which measures the fraction of the attacking path completed before the packet is identified as a replayed packet. When M is located at different locations within $F's$ transmission range, the average value of $\frac{VV'}{VD}$ indicates how successfully the neighboring monitoring can defend against the replay attack.

As nodes are uniformly distributed in the network, the probability density function that the attacker receiving the data packet from F is located in any area within $F's$ transmission range can be obtained as $\frac{1}{\pi R^2}$. As $\frac{VV'}{VD} = \frac{VM'}{VM''} = \frac{y}{1+R}$, denoting the average value of $\frac{VV'}{VD}$ as $[\frac{VV'}{VD}]_{avg}$, we have

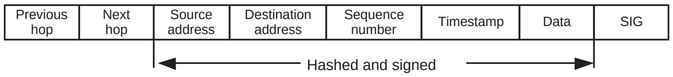
$$[\frac{VV'}{VD}]_{avg} = 2\int_{-R}^{R}\int_{0}^{\sqrt{R^2-x^2}} \frac{1}{\pi R^2}\frac{y}{y+R}dydx. \quad (1)$$

The fraction $[\frac{VV'}{VD}]_{avg}$ can then be calculated numerically, and is equal to 0.273. This means that the replayed packet can be discovered at a relatively early point of the attacking path. In particular, if the path has a hop count of no more than 3, the replayed packet can be discovered whenever the next hop of the attacker, V, forwards it on.

Neighbor monitoring can also mitigate collusive attacks. An attacker may send copies of a legitimate packet to collusive partners at different locations. Still, the colluding partners can only successfully send the replayed packet once.

Neighbor monitoring is not bullet proof. A highly mobile attacker may be able to move to a new network area in a short time, and successfully replay a packet within the packet's lifetime. A node may also use a smart antenna to prevent nodes in the neighborhood from overhearing a transmitted packet. Nevertheless, the cost and difficulty of successfully launching an attack in these cases are significantly increased.

Fig. 2 shows the proposed packet format. In the figure, the previous hop is the node forwarding the packet, and the next hop is the node designated as the receiver of the forwarded packet.



Previous hop	Next hop	Source address	Destination address	Sequence number	Timestamp	Data	SIG

Hashed and signed

Fig. 2. packet format.

If every forwarder verifies packets before forwarding them, any attack traffic will be discovered and dropped to limit its damage to the network. In particular, end servers are expected not to receive any attack packet. Network bandwidth will also be largely protected. However, verifying every packet at every forwarder causes unnecessarily high loads at the forwarders, especially when a large fraction of the packets is legitimate.

To reduce the costs of verification, without severely compromising its effectiveness, a forwarder may decide to probabilistically verify a packet. Since nodes are selfish, we need to incentivize them to verify with sufficiently high probabilities.

C. Incentives and game rationality

We apply a reward system in which nodes are given credit for acting as forwarders. Specifically, a forwarder is credited for forwarding a packet if the packet successfully arrives at the destination. We assume the existence of an accounting system, similar to a "central bank", for securely keeping track of the rewards, and preventing false rewards from being claimed. In our DoS mitigation approach, the SIG of each forwarded packet is stored at the forwarder. The stored SIGs can be presented to the accounting system as evidence for collecting rewards.

In the DoS resilient forwarding game, a node's payoff is the reward for forwarding minus the forwarding costs. The costs account for all expended resources in the forwarding, such as the energy consumed for packet receive and transmission, and for performing any required cryptographic operation.

In the DoS defense, forwarders verify the SIGs of received packets. A selfish forwarder may try to maximize its payoff by not verifying, but rely on another forwarder on the packet's route to verify and accomplish the job of filtering out any attack packet. Clearly, if every forwarder reasons in the same way and avoids all verification, then all attack packets will be allowed to reach their destinations. To avoid the degeneration of the DoS defense into a system in which no verification is performed at all, a forwarder is punished for forwarding a bad packet that successfully makes it to the destination. Hence, if a forwarder presents the SIG of a bad packet in claiming its reward, a penalty instead of a reward will be given. The penalty subtracts from the node's total credit for forwarding other good packets.

We formulate the DoS resilient packet forwarding system as a multiplayer game between forwarder nodes in a MANET. Forwarder nodes take part in the same game if they are on the same route between a sender and receiver. Since routes in a MANET can be highly dynamic, the set of nodes playing against each other can change often. As discussed, a player's payoff in the game is its reward for forwarding the good packets, less its penalty for forwarding the bad packets and its costs of forwarding and verification. A player's strategy is its probability of verifying a received packet. The player's strategy may be adaptive so that the probability of verification may change over time.

III. Game Formulation

We now formulate the formal game. We first define the reward, cost, and penalty of the game. We will then discuss a simple two-player extensive game, under both perfect and imperfect information. We will further generalize the two-player game into an n-player game in which there are n forwarder nodes along a network path. Notice that in our game formulation, we consider only the strategies of rational forwarders and how they interact with each other; in particular, an attacker is not considered a player in the game.

A. Reward, cost, and penalty

A forwarder may perform the following operations: (1) forwarding a packet without verification, (2) verifying and forwarding a legitimate packet, and (3) verifying and dropping a bad packet. Let G be the reward for a forwarder if it has forwarded a legitimate packet, and the packet is successfully delivered to the destination. Let c_p be the penalty for a forwarder if it has forwarded a bad packet without verification, and the packet reaches its destination. Let c_r, c_t, and c_v be the costs for packet receive, transmit, and signature verification, respectively.

When a forwarder forwards a legitimate packet, its payoffs are $g_1 = G - c_r - c_t$ and $g_2 = G - c_r - c_t - c_v$ for the cases of no verification and verification, respectively. If a forwarder verifies a bad packet and then drops it, the forwarder has a payoff of $g_3 = -(c_r + c_v)$. If a forwarder forwards a bad packet without verification, its payoff is either (1) $g_4 = -(c_r + c_t)$, if the packet is verified and dropped by a forwarder later in the route, or (2) $g_5 = -c_p - c_r - c_t$, if the packet finally arrives at the destination.

The different cases of payoffs for a forwarder are summarized in Tabel I.

TABLE I
Payoffs for different cases.

$g_1 = G - c_r - c_t$	a good packet without verification
$g_2 = G - c_r - c_t - c_v$	a good packet with verification
$g_3 - c_r - c_v$	a bad packet with verification
$g_4 = -c_r - c_t$	a bad packet without verification yet it is verified by following forwarders
$g_5 = -c_p - c_r - c_t$	a bad packet without verification and the packet reaches the destination

B. Two-player game

We now analyze a simple game scenario. The game involves two players, which may be the last forwarder and the second last forwarder on a route (see Fig. 3). In this game, we call the second last forwarder the *previous hop*, and the last forwarder the *next hop*.

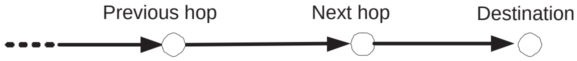


Fig. 3. A two-player game illustration.

1) **Game with perfect information:** We first consider the case that the next hop is aware of the previous hop's action. This can be achieved by having the previous hop label the packet, to indicate whether it has verified the packet's signature or not.

The game is an extensive game, and can be illustrated as in Fig. 4. We denote by V the strategy of verification before forwarding, and by F the strategy of forwarding without verification. Let G_{m-xy} be the payoff for node m (m is either the previous hop p or the next hop n), when the previous hop uses strategy x and the next hop uses strategy y. For example, G_{p-vf} is the payoff for the previous hop if it verifies the packet and the next hop forwards the packet without verification. Let p_{att} be the probability that a packet is a bad packet. The payoffs for the two players under different combinations of strategies are summarized in Table II.

We use backward induction to solve the game. Consider the subgame in which the previous hop verifies a received packet. As for a forwarder, if the payoff without verification is always higher than that with verification (i.e., $G_{n-vf} > G_{n-vv}$), then the next hop will always select forwarding (F). On the other hand, in the subgame in which the previous hop forwards the

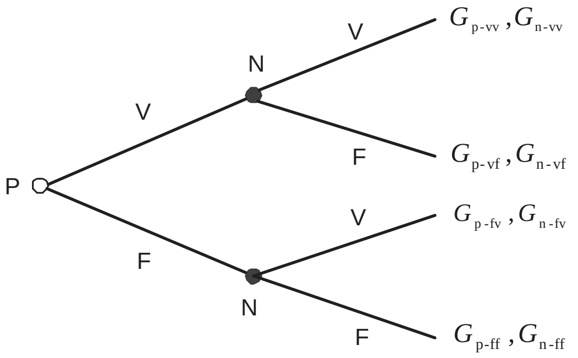

P
V
F
N
N
V
F
V
F
$G_{\text{p-vv}}, G_{\text{n-vv}}$
$G_{\text{p-vf}}, G_{\text{n-vf}}$
$G_{\text{p-fv}}, G_{\text{n-fv}}$
$G_{\text{p-ff}}, G_{\text{n-ff}}$

Fig. 4. Extensive game with perfect information.

TABLE II
EXPECTED PAYOFFS FOR THE PREVIOUS AND NEXT HOPS.

G_{p-vv}	$(1-p_{att})g_2 + p_{att}g_3$	G_{n-vv}	$(1-p_{att})g_2$
G_{p-vf}	$(1-p_{att})g_2 + p_{att}g_3$	G_{n-vf}	$(1-p_{att})g_1$
G_{p-fv}	$(1-p_{att})g_1 + p_{att}g_4$	G_{n-fv}	$(1-p_{att})g_2 + p_{att}g_3$
G_{p-ff}	$(1-p_{att})g_1 + p_{att}g_5$	G_{n-ff}	$(1-p_{att})g_1 + p_{att}g_5$

packet without verification, if $G_{n-fv} > G_{n-ff}$, then the next hop will select verification (V). Otherwise, the next hop will forward the packet. As shown in Tables I and Table II, the condition for $G_{n-fv} > G_{n-ff}$ is given as $c_v < p_{att}(c_P + c_f)$.

Now we determine what strategy the previous hop should select. When $c_v < p_{att}(c_p + c_f)$, the action pair for the next hop is FV. For the previous hop, its choice will depend on the comparison between G_{p-vf} and G_{p-fv}. As $G_{p-fv} > G_{p-vf}$ is always true in this case, the previous hop should forward the packet without verification. Hence, a Nash equilibrium strategy pair for the two players is (F, V).

When $c_v > p_{att}(c_p + c_f)$, the action pair for the next hop is FF. As $G_{p-ff} > G_{p-vf}$, the previous hop will also select F. A Nash equlibrium strategy pair for the two players is (F, F).

Whether the next hop will verify the packet or not depends on the comparison between c_v and $p_{att}(c_p + c_f)$. As c_v and c_f are fixed values, the decision depends on p_{att} and c_p. When p_{att} becomes larger, it is more likely that the next hop will verify the packet. On the other hand, if we want the next hop to verify the packet with a higher probability (e.g., the server requires that most bad packets should be discovered and dropped in the network), then the value of c_p should be larger.

The game results show that the previous hop will always forward the packet, and will leave the task of verification to the next hop. This may not be fair. In addition, from the network point of view, it is desirable for the previous hop to verify the packet, because this will limit the bandwidth wasted in transmitting the packet. To motivate the previous hop to verify, the penalty c_p can be adjusted. For any bad packet that reaches the destination, we may charge an extra penalty $c_{bandwidth}$ for the previous hop. The payoff G_{p-ff} in Tabel II is then adjusted, and now becomes $(1-p_{att})g_1 + p_{att}(g_5 - c_{bandwidth})$.

The adjusted payoff will not change the strategies of the players when $c_v < p_{att}(c_p + c_f)$. However, when $c_v > p_{att}(c_p + c_f)$, the previous hop will select V if $c_v < p_{att}(c_p + c_f + c_{bandwidth})$, and F otherwise.

2) **Game with imperfect information:** The game with perfect information leads to an "unfair" game at equilibrium since most of the work will be forced upon the next hop. In addition, a packet label of V or F by the previous hop will have to be trusted, or be verified as well, which brings additional cost.

In this subsection, we analyze the case when the next hop does not know about the previous hop's action. Without the knowledge, the next hop will have to guess whether a packet has been verified or not. For illustration, we will assume that the guess is fifty-fifty; i.e., it will have a 0.5 probability of being incorrect. The imperfect information game can be analyzed as a strategy game with a chance play. The game is illustrated in Fig. 5.

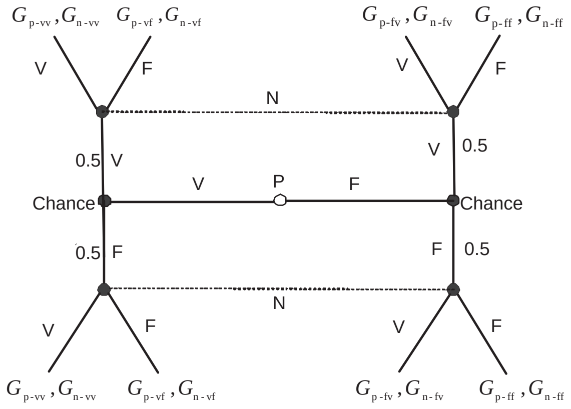

$G_{\text{p-vv}}, G_{\text{n-vv}}$
$G_{\text{p-vf}}, G_{\text{n-vf}}$
$G_{\text{p-fv}}, G_{\text{n-fv}}$
$G_{\text{p-ff}}, G_{\text{n-ff}}$
V
F
V
F
N
0.5
V
V
0.5
Chance
V
P
F
Chance
0.5
F
F
0.5
N
V
F
V
F
$G_{\text{p-vv}}, G_{\text{n-vv}}$
$G_{\text{p-vf}}, G_{\text{n-vf}}$
$G_{\text{p-fv}}, G_{\text{n-fv}}$
$G_{\text{p-ff}}, G_{\text{n-ff}}$

Fig. 5. Extensive game with imperfect information.

We denote by X/Y the case that the next hop selects strategy X based on a discovered signal of Y, where Y is the next hop's guess for the previous hop's action. For example, V/F denotes the case that the next hop verifies the packet based on its judgment that the previous hop has forwarded the packet without verification.

Let $(G_{p(Z,X/Y)}, G_{n(Z,X/Y)})$ be the payoff pair for the previous and next hops, when the next hop uses strategy X based on signal Y, while the previous hop's actual strategy is Z (where Z can be either X or Y). For example, $G_{p(V,F/V)}$ is the payoff for the previous hop when it uses strategy V, while the next hop uses strategy F upon the discovered signal V.

According to Fig. 5, the expected payoff for each player depends on their chosen strategies, and not on the signals discovered by the next hop. For example,

$$G_{p(V,V/F)} = G_{p(V,V/V)} = 0.5 * G_{p-vv} + 0.5 * G_{p-fv}; \quad (2)$$
$$G_{n(V,V/F)} = G_{n(V,V/V)} = 0.5 * G_{n-vv} + 0.5 * G_{n-fv}.$$

The strategy form for this game can thus be simplified as in Table III. The first column shows the possible strategies for the previous hop, and the first row shows the possible strategies for the next hop.

TABLE III
STRATEGIC FORM OF THE EXTENSIVE GAME WITH IMPERFECT INFORMATION.

	V	F
V	G_{p-vv}, G_{n-vv}	G_{p-vf}, G_{n-pf}
F	G_{p-fv}, G_{n-fv}	G_{p-ff}, G_{n-ff}

The extensive game is then exactly the same as a strategic game in which the two players make their moves simultaneously. Let p^* be the probability that the previous hop verifies the packet, and q^* be the probability that the next hop verifies the packet, and the mixed strategies based on p^* and q^* will lead to a Nash equilibrium of the game. Using the standard procedure [13], we can obtain p^* and q^* as follows:

$$\begin{aligned} p^* &= \frac{G_{n-ff} - G_{n-fv}}{G_{n-ff} + G_{n-vv} - G_{n-fv} - G_{n-vf}} \\ q^* &= \frac{G_{p-ff} - G_{p-vf}}{G_{p-ff} + G_{p-vv} - G_{p-fv} - G_{p-vf}} \end{aligned} \tag{3}$$

When $p^* < 0$ or $q^* < 0$, the respective best strategies of the previous and next hops are the pure strategies of forwarding without verification. Notice that in Eqn. (3), the denominators of both fractions are the same. Therefore, when p^* is 0, q^* is also 0. This means that when the network attack severity (i.e., p_{att}) or penalty (i.e., c_p) changes, both players will switch from the pure strategy F to mixed strategies simultaneously. Notice that $p^* = 0$ or $q^* = 0$ only when $c_v = p_{att}(c_p + c_t)$. Therefore, in the extensive game with imperfect information, when $c_v \geq p_{att}(c_p + c_t)$, both players will use a pure strategy. When $c_v < p_{att}(c_p + c_t)$, a mixed strategy will be used; the previous hop verifies a packet with probability p^*, and the next hop verifies a packet with probability q^*.

We have solved the case in which the guess has a 0.5 chance of being wrong. The approach in [13] can be used to solve the game under a general chance value.

C. *n*-player game

In formulating the n player game, we assume that each forwarder on a network path knows that the path has n hops. However, a forwarder does not know its position on the path; i.e., it does not know how many hops it is away from the source or the destination. In the game, each forwarder plays against the other $n-1$ forwarders. Since all the forwarders know the same information, they are treated as homogeneous and hence will use the same strategy.

We assume that upon receiving a packet, a forwarder verifies the packet with probability p_v. Nash equilibrium will be reached only if under p_v, the expected payoff for the forwarder remains the same whether it verifies the packet or not. Mathematically, the relationship can be given as follows:

$$\begin{aligned} (1-p_{att})g_2 + p_{att}g_3 = (1-p_{att})g_1 + \\ p_{att}((1-p_v)^{n-1}g_5 + (1-(1-p_v)^{n-1})g_4). \end{aligned} \tag{4}$$

The left hand side is the expected payoff when the forwarder verifies the packet. The right hand side is the expected payoff when it does not verify the packet, while the remaining forwarders will verify with probability p_v. The number of forwarders on the path is n. Based on Eqn. (4), p_v can be calculated as

$$p_v = 1 - \left(\frac{(1-p_{att})(g_2-g_1) + p_{att}(g_3-g_4)}{p_{att}(g_5-g_4)}\right)^{\frac{1}{n-1}} \tag{5}$$

The expected payoff of a player in this game can be calculated as

$$G = (1-p_{att})g_2 + p_{att}g_3. \tag{6}$$

Notice that the expected payoff of each forwarder is the same as the expected payoff if the forwarder verifies every packet. However, under the proposed game, a forwarder obtains the same gain with less consumed resources because the payoff deduction is partially caused by the penalty. This keeps the forwarders operational in the network for a longer time, by conserving nodal resources.

For comparison, we now calculate the optimum payoff for a forwarder assuming that nodes are not selfish but will collaborate for the common good. Suppose that each forwarder verifies a packet with probability p. The expected payoff for a forwarder is then

$$\begin{aligned} & G_{avg} = (1-p_{att})g_2 + p_{att}g_3 \\ + \quad & (1-p_{att})g_1 + p_{att}((1-p)^{n-1}g_5 + (1-(1-p)^{n-1})g_4). \end{aligned} \tag{7}$$

By differentiating the right hand side of Eqn. 7 and setting it equal to 0, we can solve the equation and obtain the probability of verification that will maximize the forwarder's payoff. Denoting the optimal probability by $p_{optimum}$, we have

$$p_{optimum} = 1 - \left(\frac{(1-p_{att})(g_2-g_1) + p_{att}(g_3-g_4)}{np_{att}(g_5-g_4)}\right)^{\frac{1}{n-1}}. \tag{8}$$

IV. DISCUSSIONS

We now address implementation issues for realizing the proposed signature-based DoS defense system. Our purpose is to elucidate the technical issues involved, and to propose a solution approach that is promising though preliminary. We do not claim to have resolved all the issues in the most effective/efficient manner, and further research is needed to fully characterize the system implementation and evaluate the systems tradeoffs.

A. Accounting system

An accounting system is essential for determining the rewards of forwarders and the payments by destinations. The system can operate off-line. In our system, a forwarder collects the hash results[2] of packets that it has forwarded, and present them to the accounting system. After the accounting system authenticates the claimer, it then verifies the hash results based on delivery records (provided by the packet destinations) of both the good and bad packets. Rewards or penalties are then calculated for the forwarders.

How can we prevent cheating by a forwarder who claims false credit? As it is difficult to establish secure link layer communication between any two mobile ad hoc nodes, the link layer encryption may not be used. In this case, a cheater can generate hash results even without doing the forwarding work because it can intercept packets transmitted within its receiving range. The cheating discovery method in [11] then will not work. An accounting system embedded in hardware [4] will not work either, as a cheater may choose to receive a packet and forward it, even though it is not on the network path.

We propose a statistical method to discover conventional cheaters. A conventional cheater is defined as an ad hoc node that "lives on" cheating; i.e., it claims a large amount of credits that are not deserved. In our method to prevent cheating, a forwarder keeps not only a record of the packet signatures forwarded, but also their path information including the packet's previous hop and next hop. A forwarder sends the stored path information to the accounting system. In this case, the accounting system can verify the paths of delivered data based on the reports by all the forwarders. Note that the path verification is only needed when there are conflict claims.

The basic method to prevent cheating is vulnerable to a collusive attack as shown in Fig. 6. In the figure, Attacker 1 records its collusive partner, Attacker 2, as the next hop, instead of recording the real next hop. If the partner is close enough to the real next hop, it can identify the node to which the next hop forwards the packet. The partner will then be able to claim (false) credit for the forwarding.

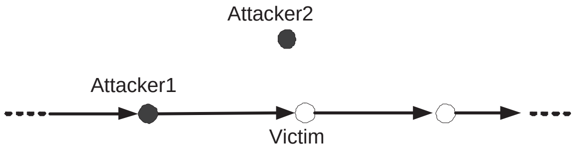


Fig. 6. A collusive attack on the accounting system to claim false credits.

A defense against the collusion is as follows. Whenever the accounting system has more than one node showing the same path payment information, it will hold the payment and, at the same time, keeps the nodes in a *questionable list*. Statistically, attackers may cheat repeatedly. Therefore, the frequency at which a cheater is added to the questionable list will be significantly higher than that of a legitimate node. The accounting system can then use the frequency information to identify a cheater with high accuracy. After identifying the attacker, the accounting system pays the legitimate forwarders.

B. Setting game parameters

The costs of packet send, receive, and signature verification should be set based on the resources consumed. Among the resources, energy is critical because unlike other resources such as storage and CPU cycles, energy of mobile nodes can be difficult or impossible to replenish. Therefore, energy consumption can be a deciding factor for determining the costs of network operations. Similarly, the relative costs of send, receive, and signature verification can be compared based on their energy consumption [12], [18].

The reward for each forwarder after the successful delivery of a good packet should depend on the benefits the delivery brings to the source or destination. It may also be determined by the network's desire to encourage a node to act as a forwarder. In this paper, we set the reward for each forwarder as a constant value, regardless of the number of forwarders on the path. The decision is reasonable because more network resources are consumed when there are more forwarders, hence justifying a higher payment by the destination.

The value of penalty can be determined based on damage to the server or network when a bad packet reaches the target. The penalty can also be used by the accounting system to affect the willingness of a forwarder to verify packets. To make the game more efficient, we propose to set the penalty as a function of p_{att}, so that when p_{att} is higher, the penalty value is higher. This is because the more serious the attack, the more we would like a forwarder to verify a packet. We study the use of a linear and an exponential function, to express the relationship between the penalty and the severity of attack.

When using a linear function, the penalty can be calculated as:

$$c_p = c_1 + kp_{att}. \tag{9}$$

When using an exponential function, the penalty can be calculated as:

$$c_p = c_1 e^{kp_{att}}. \tag{10}$$

These two functions are applied by the system to control the behaviors of individual nodes. The goal is to have a high enough penalty to reduce the number of bad packets received by a server to an acceptable level. However, the penalty value should not be too high. Otherwise, network nodes will not be motivated to act as forwarders, because they cannot obtain a sufficient payoff for doing so. As a special case, when k equals 0, the penalties computed by both of the equations are constants.

V. Numerical Results

We quantify the costs of packet send, receive, and verification as their amounts of energy consumption. We use the experimental values reported in [12], [18]. We set the length of a packet as 1000 bytes. We normalize the costs by the power

[2] A forwarder generates the hash results upon receiving a packet.

consumption of a packet receive (i.e., one packet receive has cost one). The costs of send and verification (including hashing and RSA verification) are then 1.5 and 10, respectively. For illustration, we set the reward and penalty values to be 20 and 60, respectively. In future work, these costs will have to be refined to model specific systems. Where applicable, the extra penalty (see Section III-B1) for forwarding a bad packet, $c_{bandwidth}$, is set to be 3. The number of forwarders on a path, n, is set to be 3. Unless otherwise specified, $p_{att} = 0.2$.

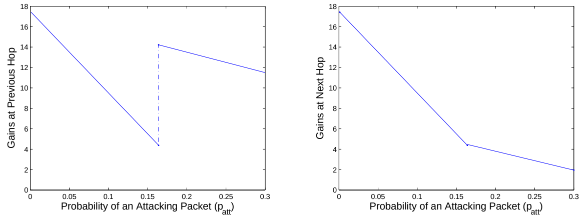
The baseline values of the game parameters are summarized in Table IV.

TABLE IV
GAME PARAMETER SETTINGS

c_r	c_t	c_v	G	c_p	n	p_{att}	$c_{bandwidth}$
1	1.5	10	20	60	4	0.2	3

A. Two-player game

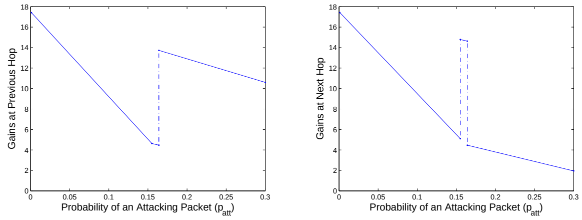
We first examine the game with perfect information. In Fig. 7, we show the payoffs for the previous and next hops, respectively, as the probability of a bad packet p_{att} changes. When p_{att} is small, the payoffs for both players decrease as p_{att} increases. When p_{att} reaches the threshold point of $\frac{c_v}{c_p+c_f} = 0.1639$, the payoff for the previous hop will jump to a high value because the second hop switches its strategy from F to V. After the threshold point, both payoffs again decrease as p_{att} increases. In Fig. 8, we show the payoffs when network loss is considered. There are two threshold values: $p_{att} = 0.1639$ and $\frac{c_v}{c_p+c_f+c_{bandwidth}} = 0.1550$. When $p_{att} = 0.1550$, there is a large increase in the next hop's payoff because the previous hop changes its strategy from F to V. At $p_{att} = 0.1639$, the payoff for the previous hop has a large increase while, at the same time, the payoff for the next hop decreases significantly. The reason is that the previous hop changes its strategy from V to F, while the next hop changes its strategy from F to V. In both games, the probability that a packet received by the destination is a bad packet is p_{att} if both hops use strategy F, and 0 if any of them uses strategy V.



(a) Previous hop

(b) Next hop

Fig. 7. Previous hop and next hop payoffs with perfect information.



(a) Previous hop

(b) Next hop

Fig. 8. Payoffs with perfect information: The case of extra penalty.

In Fig. 9 we show, for a game with imperfect information at Nash equilibrium, the probabilities that the previous hop and next hop will verify a packet, respectively. When p_{att} is low, both of the players will use the pure strategy F because it is dominant. When $p_{att} \geq 0.1639$, both players switch to a mixed strategy. The probability that a player verifies the packet increases when the attack becomes more severe. In Fig. 10, we show the expected payoffs for the two players. Both payoffs decrease as the attack gets more severe.

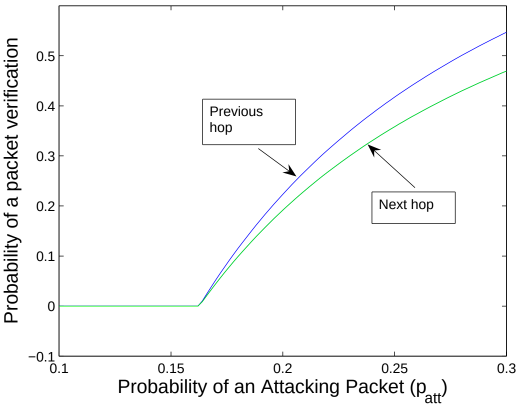


Fig. 9. Probabilities of verification under different percentages of bad packets.

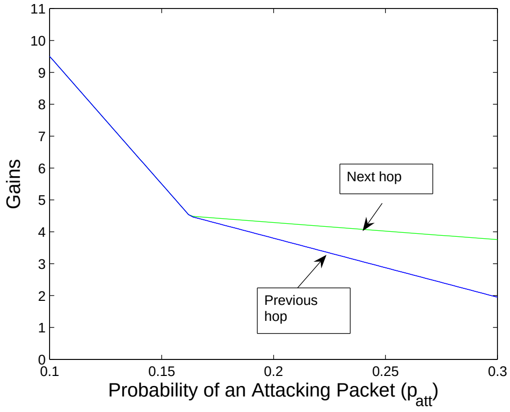


Fig. 10. Payoffs under different percentages of bad packets.

B. *n*-player game

Fig. 11 shows that when p_{att} increases, a node has to verify packets more often in order to reach Nash equilibrium. Note that a player will use the mixed strategy when $p_{att} \geq 0.1639$, and the probability of verification increases as the attack gets more severe.

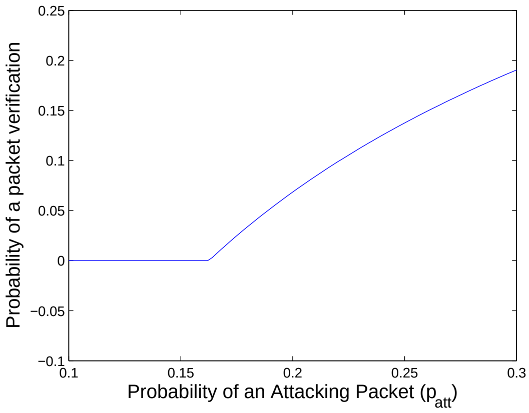


Fig. 11. Probabilities of verification at Nash equilibrium.

In Fig. 12, we show the payoff for a forwarder as p_{att} changes, and in Fig. 13, we show the probabilities that a bad packet can reach the destination. For comparison, we also show the cases (1) when the best strategies are used under the assumption that the nodes are collaborative, and (2) when the worst strategy is used so that all the nodes simply forward packets without any verification.

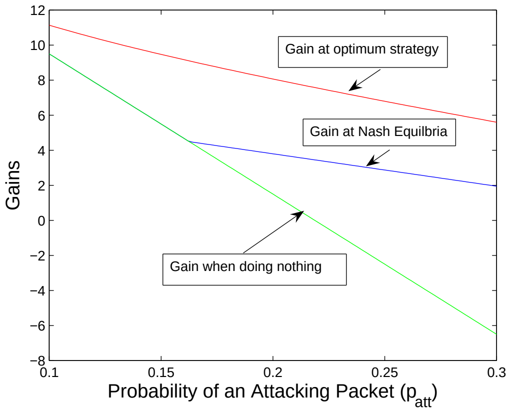


Fig. 12. Payoffs at Nash equilibrium.

Fig. 14 shows how a forwarder may change its strategy as the penalty value changes. As the penalty becomes high, a forwarder will more likely verify the packet to avoid a negative payoff. Note that when the probability of receiving a bad packet is the same, the expected payoff at Nash equilibrium for any player remains the same even if the penalty value changes. In Fig. 15, we show the probability that a bad packet will successfully arrive at the destination. This probability decreases as the penalty value increases.

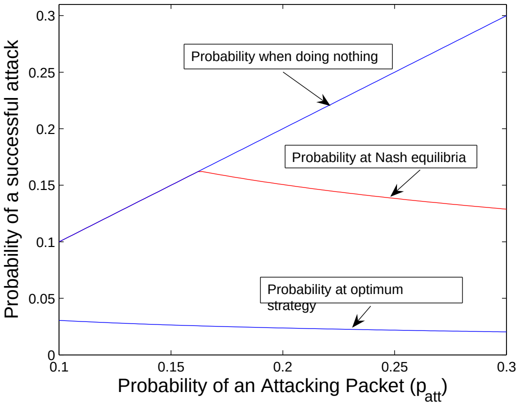


Fig. 13. Probabilities that a bad packet will successful arrive at its destination.

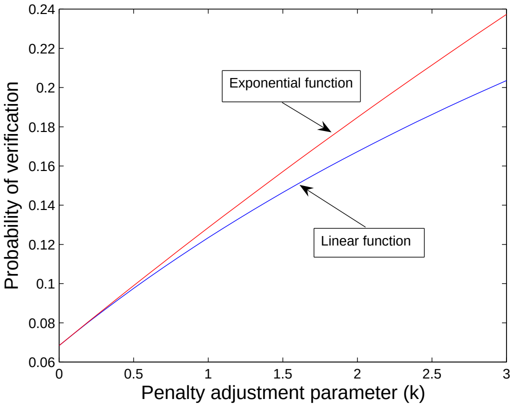


Fig. 14. Impact of penalty on decision of a forwarder.

C. Summary of observations

Based on the numerical results, we summarize the important observations as follows:

- In a two player game, the game with perfect information to the numerical results. will generate higher payoffs for both players. However, the perfect information will cause the next hop to do all the verification work. In contrast, the game with imperfect information can cause the previous hop to share some of the verification work, which is desirable for conservation of global network bandwidth.
- Players (i.e., the forwarders) will change their strategies when p_{att} reaches certain threshold values. This is because when p_{att} is low, the gain from verification is not enough to compensate for its costs. It is interesting that in all the games studied in this paper, the threshold value of p_{att} at which a player will change its strategy is the same and is equal to $\frac{c_v}{c_p+c_f}$. The common threshold value is a function of the forwarding cost, the verification cost, and the penalty. The costs of forwarding and verification should be determined by the resource consumption of the

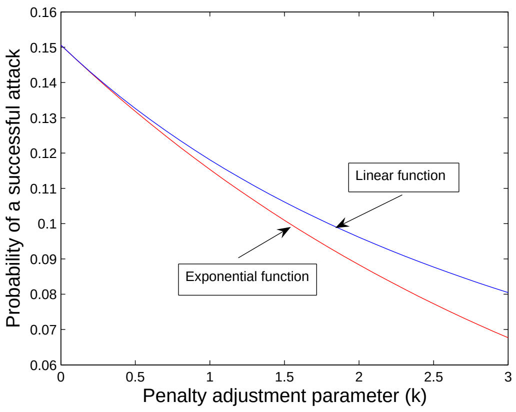


Fig. 15. Impact of penalty on DoS mitigation.

operations, instead of by policy. The tunable parameter to obtain certain desired game results is therefore the penalty value. For example, to motivate a forwarder to verify more often, we can use a larger penalty value.

- In the n-player game, the payoff for a forwarder decreases as the attack becomes more severe. The proposed game helps mitigate DoS because the probability for a bad packet to reach its destination decreases even as the attack becomes worse. This indicates that the effectiveness of the DoS mitigation game increases as the attack becomes more severe.
- In the n-player game, the payoff for each forwarder is fixed (see Eqn. (6)). When the penalty value is high enough, the probability for a bad packet to reach the destination can be reduced to a small value. However, the resources (e.g., nodal energy) consumed at a forwarder will be more, and hence the nodes will have a shorter lifetime if the resources are not replenishable. Therefore, the penalty value should be properly selected to keep the rate of successful attack low and at the mean time, to motivate forwarders.

VI. Conclusions and Future Work

We have proposed a signature-based DoS mitigation system for mobile ad hoc networks. The system defines a game in which forwarders will probabilistically verify packets received for forwarding, and hence will have a chance to drop bad packets sent by attackers. We have formulated different forms of the game for different network scenarios, and analyzed the corresponding payoff, effectiveness, and Nash equilibrium properties. Based on our analysis and numerical results, it can be concluded that the games can induce useful DoS mitigation effects. In addition, key game parameters, such as the penalty for forwarding a bad packet without verification, can affect the probability that a node will verify a received packet. Therefore, determining the value for the penalty can be effective for game control.

Future work includes experiments under more involved operating scenarios, and the design of protocols to accurately estimate the severity of attack. It would also be interesting to investigate the existence and attainment of Nash equilibrium when a forwarder does not know the length of the network path. Finally, issues of determining realistic reward and penalty values under specific deployment scenarios should be studied.

VII. Acknowledgement

Research was supported in part by the U.S. National Science Foundation under grant number CNS-0305496, and in part by an Asian Initiative grant awarded by Purdue University.

References

[1] T. Alpcan and T. Basar. A game theoretic approach to decision and analysis in network intrusion detection. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003.

[2] K. Argyraki and D. R. Cheriton. Active internet traffic filtering: Real-time response to denial-of-service attacks. In *In USENIX Annual Technical Conference*, 2005.

[3] S. Buchegger and J.-Y. L. Boudec. Performance analysis of the confidant protocol. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, 2002.

[4] L. Buttyan and J. Hubaux. Nuglets: a virtual currency to stimulate cooperation in self-organized ad hoc networks. In *Technical report, EPFL*, 2001.

[5] M. Felegyhazi and J.-P. Hubaux. Game theory in wireless networks: A tutorial. *EPFL technical report*, (LCA-REPORT-2006-002), 2006.

[6] M. Felegyhazi, J.-P. Hubaux, and L. Buttyan. Nash equilibria of packet forwarding strategies in wireless ad hoc networks. *IEEE Transactions on Mobile Computing*, 5, 2006.

[7] Y.-C. Hu, D. B. Johnson, and A. Perrig. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *Proceedings of Mobicom*, 2002.

[8] J. Ioannidis and S. M. Bellovin. Implementing pushback: Router-based defense against ddos. In *In Proceedings of Network and Distributed System Security Symposium*, 2002.

[9] B. Karp and H. T. Kung. Gpsr: Greedy perimeters stateless routing for wireless network. In *Proceedings of ACM/IEEE Mobicom*, pages 243–254, 2000.

[10] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *Proceedings of ICNP*, 2001.

[11] H. Luo, R. Ramjee, P. Sinha, L. Li, and S. Lu. Ucan: a unified cellular and ad-hoc network architecture. In *Proceedings of the 9th Mobicom*, 2003.

[12] L. M. Marie and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *Proceedings of IEEE INFOCOM*, 2001.

[13] M. J. Osborne. *An Introduction to Game Theory*. Oxford University Press, Inc., 2004.

[14] K. Romer. Time synchronization in ad hoc networks. 2001.

[15] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for ip traceback. In *Proc. of 16th ACM SIGCOMM (SIGCOMM'2000)*, 2000.

[16] A. Snoeren, C. Partridge, L. Sanchez, and C. Jones. Hash-based ip traceback. 2001.

[17] V. Srinivasan, P. Nuggehalli, C. F. Chiarresini, and R. R. Rao. Cooperation in wireless ad hoc networks. In *Proceedings of Infocom*, pages 808–816, 2003.

[18] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications*, 2005.

[19] A. Yaar, A. Perrig, and D. Song. Siff: A stateless internet flow filter to mitigate ddos flooding attacks. In *In Proceedings of the IEEE Security and Privacy Symposium*, 2004.

[20] D. W. X. Yang and T. Anderson. A dos-limiting network architecture. In *In Proc. ACM SIGCOMM*, 2005.

[21] S. Zhong, L. E. Li, Y. G. Liu, and Y. R. Yang. On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an integrated approach using game theoretical and cryptographic techniques. In *MobiCom '05: Proceedings of the 11th annual international conference on Mobile computing and networking*, 2005.