COLLABORATION ENFORCEMENT AND ADAPTIVE DATA REDIRECTION IN MOBILE AD HOC NETWORKS USING ONLY FIRST-HAND EXPERIENCE

Ning Jiang, Kien A. Hua, Mounir A. Tantaoui
{njiang, kienhua, tantaoui}@cs.ucf.edu
School of Computer Science
University of Central Florida
Orlando, FL 32816-2362

Abstract: **In *Mobile Ad Hoc Networks* (MANETs), all participating hosts are obligated to route and forward data for others to guarantee the availability of network applications and services. Most of the contemporary collaboration enforcement techniques employ reputation mechanisms for nodes to avoid and penalize malicious participants. Reputation information is updated based on complicated trust relationships among hosts and other techniques to thwart false accusation of benign nodes. The aforementioned strategy suffers from low scalability and is likely to be exploited by adversaries. In this paper, we propose a novel approach to address the aforementioned problems. With the proposed technique, no reputation information is propagated in the network and malicious nodes cannot cause false penalty to benign hosts. Misbehaving nodes are penalized and circumvented by benign nodes within their localities based on first-hand experiences. This approach significantly simplifies the collaboration enforcement process, incurs very low overhead, and is robust against various evasive behaviors. Simulations based on various system configurations demonstrate that overall network performance is greatly enhanced.**

Key words: Mobile Ad Hoc Network, Collaboration enforcement, Reputation, First-hand experience

1. INTRODUCTION

Mobile Ad hoc NETworks (MANETs) has attracted great research interest in recent years. A Mobile Ad Hoc Network is a self-organizing multi-hop wireless network where all hosts (often called nodes) participate in the routing and data forwarding process. The dependence on nodes to relay data packets for others makes mobile ad hoc networks extremely susceptible to various malicious and selfish behaviors. This point is largely overlooked during the early stage of MANET research. Many works simply assume nodes are inherently cooperative and benign. However, experiences from the wired world manifest that the reverse is usually true; and many works [3] [10] [9] [8] [12] [19] have pointed out that the impact of malicious and selfish users must be carefully investigated. The goal of this research is to address the cooperation problem and related security issues in wireless ad hoc networks. As a rule of thumb, it is more desirable to include security mechanisms in the design phase rather than continually patching the system for security breaches.

As pointed out in [2] [1], there can be both selfish and malicious nodes in a mobile ad hoc network. Selfish nodes are most concerned about their energy consumption and intentionally drop packets to save power. The purpose of malicious nodes, on the other hand, is to attack the network using various intrusive techniques. In general, nodes in an ad hoc network can exhibit Byzantine behaviors. That is, they can drop, modify, or misroute data packets. As a result, the availability and robustness of the network are severely compromised. A common solution to combat such problems is for each node to maintain a reputation list of other nodes, as proposed in [1] [3][13][14]. In these techniques, misbehaving nodes are detected and a rating algorithm is employed to avoid and penalize them. These schemes are not scalable and suffer from high overhead since they require synchronization of reputation information throughout the network, and manipulation of complicated trust relationships among hosts to thwart false accusation of benign nodes. In this paper, we propose a novel approach to strengthening collaboration in MANETs. With this scheme, no reputation information needs to be propagated in the network and malicious nodes cannot cause false penalty to benign hosts. Misbehaving nodes are penalized and circumvented by benign nodes within their localities based on first-hand experiences. This approach significantly simplifies the collaboration enforcement process, incurs very low overhead, and is robust against various evasive behaviors. Simulations based on various system configurations demonstrate that overall network performance is greatly enhanced.

The remainder of this paper is organized as follows. We discuss related works in Section 2. In Section 3, we introduce the selfish/malicious node detection mechanism, and also present the proactive rerouting techniques. Experimental results are given in Section 4. Finally, we conclude the paper in Section 5.

2. RELATED WORK

The current state of the art in enforcing collaboration in mobile ad hoc networks can be categorized into two groups, namely incentive motivation approaches and misbehavior penalty approaches.

The incentive motivation techniques are discussed in [5] [6] [7] [20]. These techniques either rely on tamper proof security modules or assume a central control service. The practicability and performance remain unclear.

Our research, on the other hand, falls in the second category. The main idea is to detect and penalize malicious and selfish behaviors. In [18], the authors use intrusion detection techniques to locate misbehaving nodes. A watchdog and a path rater approach is proposed in [13] to detect and circumvent selfish nodes. The main drawback of this approach is that it does not punish malicious nodes. This problem is addressed in [2] [3] [4]. The approach, called CONFIDANT, introduces a reputation system whereby each node keeps a list of the reputations of others. Malicious and selfish nodes are detected and reputation information is propagated to "friend" nodes, which update their reputation lists based on certain trust relationships. During route discovery, nodes try to avoid routes that contain nodes with bad reputations. Meanwhile, no data forwarding service is provided for low reputation nodes as a punishment. Another reputation-based technique, called CORE, is proposed in [14]. In [1], the authors attack the problem of defending application data transmission against Byzantine errors. In their approach, each node maintains a weight list of other nodes. Malicious nodes are located by an on-demand detection process and their weights are increased consequently. A routing protocol is designed to select the least-weight path between two nodes. This approach is also based on per-node reputation lists. In addition, the detection process requires that each intermediate node transmit an acknowledgement packet to the source node.

In general, most of existing detection and reaction techniques are based on global reputation mechanism and suffer from the following drawbacks. First, global reputation schemes have low scalability. Significant overhead is needed to propagate reputation information for all the benign nodes to avoid and punish "bad" citizens. Likewise, considerable efforts need to be made for malicious or falsely accused nodes to rejoin the network. Second, global reputation schemes offer incentives to various attacks. Most prominently, malicious users can "poison" the reputation lists by disseminating incorrect reputation information. Such packets can be spoofed with other nodes' addresses to hide the identity of the attacker or to pretend to be a "friend" of the receiver. In [1], digital signatures and message authentication codes [22] are employed to defeat packet spoofing. However, if a host is possessed (or physically captured) by a malicious user, cryptographic information of the particular node can be extracted and reputation poison attacks can still be mounted.

We propose a technique to address all the aforementioned problems by using only first-hand experience at each individual node instead of relying on globally propagated reputation. This strategy is both effective and efficient.

3. THE EXPERIENCE-BASED APPROACH

In this section, we introduce the proposed experience-based techniques. Our approach is based on the following fundamental characteristics of MANETs:

- Each packet transmitted by a node *A* to a destination node more than one hop away must go through one of *A*'s neighboring nodes.
- *A*'s neighboring nodes can overhear its packet transmission.

Given a selfish node *M*, its un-collaborative behavior can be captured by most, if not all, of its neighboring nodes. Each of these nodes will then penalize *M* by rejecting all its packets. As a result, *M* will not be able to send any data to nodes more than one hop away. For a benign node *B*, if *B* is relaying packets for a source node *S* and is aware that the next hop node *H* is a selfish node, *B* can redirect the packets to avoid *H*. Note that the rerouting operation requires collaboration from *B* for *S*. We also present techniques to enforce such collaboration.

3.1 Node Configurations

The proposed technique is based on nodes with the following configuration. First, nodes are equipped with omni-directional antennas and wireless interface cards that can be switched to promiscuous mode to "hear" data transmission in their proximities. Second, we base our discussion on the Dynamic Source Routing protocol [11], as it is one of the most frequently used routing protocols in the literature. However, the technique can be extended to accommodate other reactive routing protocols. Overview of DSR is omitted in the interest of space. Third, 802.11 [21] is employed at the MAC layer. Finally, nodes have knowledge of their one-hop neighboring nodes. This can be achieved by either employing a HELLO protocol or by overhearing packets transmitted within the locality.

3.2 Selfish and Malicious Behaviors Considered

A selfish node can avoid the responsibility of forwarding data in two ways. First, by not participating in route discovery, a node will never show up in any routing control packets and will thus be completely released from forwarding data packets. Second, a selfish host can cooperate in route discovery, but subsequently discards data packets to save energy. We focus on the second type of selfishness in this paper as it is pointed out in [16] that such misbehavior has more negative impact on overall network throughput.

3.3 Detection and Punishment of Selfishness and Malice in Data Forwarding

In the proposed technique, each node maintains a list of its neighboring nodes and tracks their actions. Nodes make no assumption of other hosts beyond their direct observable regions.

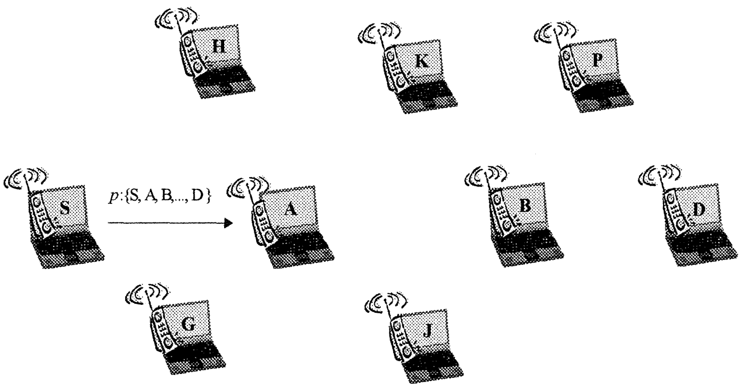

Figure 1. Detection example

We discuss the detection mechanism through an example depicted in Figure 1. It shows a node S transmitting data to a node D using a route $\{S, A, B, D\}$. Node A is a selfish node that does not forward the data packets to save energy. Assume nodes H and G are neighboring to both S and A, and Nodes K and J are neighboring nodes of both A and B. Each node allocates some memory buffer to store packets transmitted by its neighboring nodes. Let us consider node S first. After S transmits a data packet to A, it waits for a certain time interval and validates whether A has properly forwarded the packet by checking its memory buffer. Given the current validation time t, S

maintains a set Ω of all the packets it has transmitted over a time window defined by $[t - W_{UPPER}, t - W_{LOWER}]$. If the cardinality of Ω is greater than a threshold T_{SUM}, S computes the packet drop ratio for node A on Ω. If this ratio is beyond a given threshold $T_{SELFISH}$, then S tallies A as a selfish node; otherwise, S deems A as benign. The proposed detection procedure distinguishes link breakage and temporary network congestion from deliberate packet discarding, and effectively reduces false classifications. Essentially, selfish intention is sustained if and only if a node has been observed to drop a significant number of packets over a long enough timeframe. We now consider nodes *G* and *H*. They, as neighboring nodes of *S*, overhear all data packets sent by *S* and can learn about the next hop (*A* in this example) of each data packet *p* by extracting the source route option field of *p*'s IP header. As *G* and *H* are both neighboring to *A* and since *S* is a benign node, *G* and *H* will further detect whether *A* relays the packet using the same technique used in *S*. In this example, both *G* and *H* will eventually identify *A* as a selfish node based on their own observations. On the other hand, although *K* and *J* are also neighbors of node *A*, they will not be able to detect *A*'s misbehavior since they have no access to the packets sent by the previous hop to *A* (*S* in this example). We refer to this scenario as "*asymmetric sensing.*" In practice, *A* is likely to receive packets from all directions and will eventually be captured by all its neighbors. We note that users are motivated to monitor their locality as they will benefit from identifying and circumventing selfish neighboring nodes. This detection mechanism fits naturally into DSR as in DSR nodes constantly sense the media and extract routes from overheard packets. No extra energy cost is introduced by the proposed technique.

One major advantage of the proposed technique is that colluding is not an issue. In money-incentive models, significant effort needs to be invested to prevent participants from gaining monetary benefit through colluding. In reputation-based schemes, colluding is attractive to both selfish and malicious users. On one hand, colluding selfish users can successfully cover each other and escape penalty. On the other hand, malicious participants can collaboratively cause various undesirable effects to benign users. Since the proposed technique is based on direct experience at individual nodes, not through "rumor" or "propagated information," colluding is not possible in this new environment.

Punishment is enforced as follows. Consider a node *H*, which identifies node *A* as a selfish or malicious node, it will reject data packets originated by *A* for a period of time as a penalty. More specifically, *H*'s decision on whether to forward a data packet *p* for *A* is based on the difference between the time *H* receives *p* and the latest recorded time when *A* was identified as a misbehaving node. If this difference falls within a threshold defined as *penalty interval* τ, *H* will reject the packet. Consequently, the penalty will not end as long as *A* continues to misbehave, and the actual penalty time is proportional to the length of *A*'s misbehavior.

3.4 Dynamic Redirection

In global reputation mechanism, two scenarios will cause a source node to reroute data packets over a particular node. First, when a node detects a selfish or malicious node, it informs other nodes (including the source node of the session) through *reputation packets* so that they can choose a "clean" route to circumvent the selfish node. Second, Route Error (RERR) packets are transmitted to the source node when broken links are encountered[1]. In both cases, source nodes are responsible for rerouting the data. In the proposed technique, we allow neither of the above packets to be propagated. An obvious question is: who should reroute the data packets to bypass both irresponsible nodes and broken links?

Our solution is that each node shares the responsibility of rerouting packets. Again, we use Figure 1 to illustrate the idea. We assume that node *S* is sending data to node *D* through a path {*S*, *A*, *B*, *D*}. Suppose the link between node *A* and node *B* is a *malfunction link* (i.e. either broken or node *B* is selfish). Without loss of generality, we assume that node *B* is a selfish node. After relaying a certain number of packets, node *A* will realize that *B* is a selfish node. We refer to node *A* as a *proxy* of source node *S*[2]. In our approach, *A* first purges all paths containing node *B* as an intermediate node from its route cache. Next, when *A* receives subsequent data packets from *S*, it broadcasts a Route Redirect (RRDIR) packet, indicating node *B* as a *bypassing target* and then reroutes the packets by obtaining an alternative clean route to node *D* from its route cache. If such a route does not exist in its cache, *A* will buffer the data packets and instantiate a route discovery process to locate a path to *D*. In Figure 1, *A* will discover a new route {*K*, *P*, *D*}, revise the embedded route of each data packet and relay them to the destination. In this case, the actual route data packets traverse from *S* to *D* is {*S*, *A*, *K*, *P*, *D*}. It is possible for several proxy nodes to adaptively reroute data packets to avoid multiple selfish nodes along the chosen route. If *A* cannot find a route to *D* after a certain number of retries, it informs *S* through a RERR packet.

The proper functioning of the proposed selfish and malicious node circumvention scheme relies on the collaboration of proxy nodes. Unfortunately, proxy nodes can act maliciously to either avoid the reroute

[1] Selfish nodes can falsely claim broken links to be excluded from packet transmission sessions.

[2] The proxy of a source node can be the source node itself when its next hop is selfish.

task or mount denial of service attacks. Continue the above example. When node *A* receives a data packet from *S*, it has the following choices.

- Node *A* can mount a denial of service attack to *S* by deliberately forwarding packets to *B* even though it is aware that *B* is a selfish node. Nodes *K* and *J* will detect such attack as follows. First, both nodes will identify node *B* as a misbehaving node and they will assume that *A* has reached the same conclusion. Next, as *A* makes no effort to bypass *B*, both *K* and *J* will mark *A* as a malicious node and starts to penalize it.
- *A* does not reroute the packet and simply reports a RERR back to the source. In this case, all its neighboring nodes (*S*, *G*, *H*, *J*, and *K*) hear the RERR packets whereas none of them is aware of any route discovery attempt made by *A*. Thus, all of them will deem *A* as a selfish node.
- *A* broadcasts a RRDIR packet and then starts a route discovery process. Nevertheless, *A* reports a RERR to the source regardless of whether it receives RREP packets from the destination. The countermeasure we design involves utilizing some context information. After *A* sends a RREQ packet to look for a route to *D*, all its neighboring nodes will wait for the RREP packet to come back. Suppose node *K* relays the replying RREP packet to *A* and assume node *H* also hears the packet. Both *H* and *K* will expect to see node *A* transmit data to node *D*. However, as *A* sends a RERR packet, both nodes will recognize *A* as misbehaving. Furthermore, other neighboring nodes (*S*, *G*, and *J*) will deduct certain number of points for node *A* (say, equivalent to one third of those deducted for packet dropping). In other words, failure to reroute data packets is deemed as low-weight misbehavior. The purpose of this design is to discourage un-collaborative behavior. Benign nodes always collaborate and will not suffer from such deduction.
- *A* broadcasts a RRDIR packet and reroutes data through a fabricated path. This attack has very limited effect in that benign nodes along the faked route will reroute the data packets and node *A* still has to relay data.

Another concern is that malicious nodes might exploit the reroute mechanism to disrupt data transmission. For instance, in Figure 1, suppose *A* is a malicious node. When it receives a data packet that it should forward to a benign node *B*, it redirects the packet to a different (fabricated) route, hoping that other nodes along the redirected route will drop the packet. With our redirection mechanism, *A* has to broadcast a RRDIR packet. Otherwise its neighboring nodes (*S*, *H*, and *G*) will identify it as a malicious node. In the RRDIR packet, *A* has to declare the correct next hop (*B* in this case) that it intends to bypass. Otherwise, it will be captured by *S*, *H*, and *G*. After receiving *A*'s RRDIR packet, node *B* will realize *A*'s attempt to deviate packets from a valid route and penalize *A*. Nodes *K* and *J* will also penalize *A* as they both recognize *B* as a benign node through their own experiences.

Moreover, nodes that reroute packets for an excessive number of sessions within a certain time period will be considered as malicious and penalized by their neighbors.

4. EXPERIMENTAL STUDY

We implemented four schemes, namely the reference scheme, the defenseless scheme, the reputation-based scheme and the proposed experience-based scheme, for performance evaluation. In the reference scheme, all the nodes act collaboratively and relay data for each other. In the defenseless scheme, a certain fraction of nodes are selfish as they forward routing packets, but discard any data packet not destined at them. No detection or prevention mechanism is implemented so that the network is totally "defenseless". Next, we implemented a reputation-based system. In this scheme, each node maintains global reputation of other nodes. Nodes update reputation of others as follows. First, nodes monitor and form their opinion about the reputation of neighboring nodes. Nodes always trust their first-hand experiences with other nodes and ignore any reputation information against their own belief. Next, when a node detects a selfish node, it informs the source node of the communication session through a *reputation packet*. Finally, each node periodically broadcasts reputation of other nodes in its locality. We implemented three types of nodes in this scheme, namely benign node, selfish node, and cheating node. A benign node always truthfully broadcasts the reputation information it has observed first hand, and honestly forwards the reputation packets. A selfish node does not participate in data packet forwarding but cooperates in disseminating reputation information (i.e. it generates and relays reputation packets and never lies about other nodes). A cheating node generates genuine reputation packets and relays both data and reputation packets for others. During reputation broadcast, however, it always lies about the reputation of its neighboring nodes. For all other nodes it is aware of, the cheating node simply reports them as selfish.

We performed all the experiments based on GlomoSim [17], a packet-level simulation package for wireless ad hoc networks. Our experiments were based on a mobile ad hoc network with 50 nodes within a 700x700-square-meter 2-dimensional space. The simulation duration for each run was 10 minutes. The random waypoint model was used to model host mobility. We experimented with 5 and 10 selfish nodes, accounting for 10% and 20% of total number of nodes, respectively. Selfish nodes are randomly generated for all the simulation schemes. We tested the reputation-based system with 0 and 5 randomly selected cheating nodes. Each configuration was executed under 5 different random seeds and the average values of the metric variables were calculated. Constant Bit Rate (CBR) applications were used in this study. For each simulation run, we randomly generated a total of 10 CBR client/server sessions. In particular, we generated three *selfish sessions* (i.e. sessions originated by selfish nodes) and seven *benign sessions* (i.e. sessions started by benign nodes).

In the experiments, we evaluated the proposed scheme based on the goodput of benign sessions and selfish sessions. A good collaboration enforcement technique should ensure a high benign session goodput as well as suppressing selfish session goodput to discourage misbehaviors.

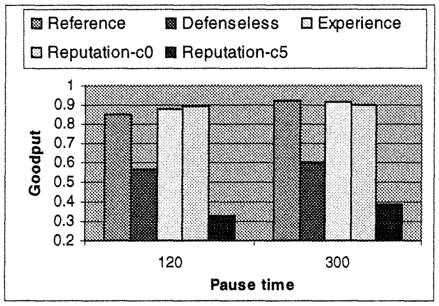

Figure 2. Benign session goodput (m=10)

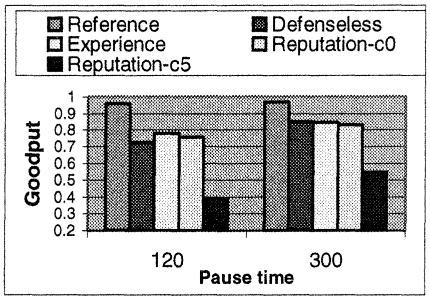

Figure 3. Selfish session goodput (m=10)

Figure 2 and Figure 3 illustrate the goodput of the experimented schemes when there are ten selfish nodes. In both figures, we refer to the proposed scheme as "Experience," and the reputation-based scheme as "*Reputation-cX*", where *X* indicates the number of cheating nodes. From Figure 2 we observe that by employing the proposed scheme, significantly more data are successfully delivered to the destination nodes since proxy nodes proactively detect and reroute data around misbehaving nodes. The proposed technique lifted the goodput from around 0.6 in a defenseless network to higher than 0.85, an improvement of more than 40%. From Figure 3, we observe that the goodput experienced by selfish users is lower than what collaborative users enjoy. When the pause time is 300 second the goodput of benign sessions is approximately 0.92 (Figure 2) as opposed to 0.81 in the case of selfish sessions (Figure 3). Finally, in all the experiments, the reputation-based scheme suffered from significant performance loss (more than 50%) when only a few cheating nodes were present. We thus conclude that experience-based scheme is more suitable for MANETs due to its resilience to performance degradation caused by reputation poisoning behaviors.

5. CONCLUDING REMARKS

In mobile ad hoc networks, there is no fixed infrastructure readily available to relay packets. Instead, nodes are obligated to cooperate in routing and forwarding packets. However, it might be advantageous for some nodes not to collaborate for reasons such as saving power and launching denial of service attacks. Therefore, enforcing collaboration is essential in mobile ad hoc networks.

In most existing techniques, collaboration enforcement is achieved by a detect-and-react mechanism. In which, each node maintains global reputation of others in order to avoid and penalize misbehaving nodes. Propagation of reputation information is accomplished through complicated trust relationships. Such techniques incur scalability problems and are vulnerable to various reputation poisoning attacks.

In this paper, we proposed a novel approach to enforcing collaboration and security in mobile ad hoc networks. In our technique, nodes keep local reputation of their neighboring nodes through direct observation. No reputation advertisement is initiated or accepted. Nodes dynamically redirect data packets to avoid recognized adversaries. The redirect operation is also guarded against various evasive attempts. The advantages of this approach are many. First, since it does not rely on propagated reputation information, there is no need to maintain complex trust relationships. Second, since the misbehavior detection mechanism is based on first-hand experience at individual nodes, denial of service attacks are much more difficult to achieve. Colluding among nodes to secretly carry out fraudulent acts is not possible either.

We conducted various experiments to investigate the effectiveness and efficiency of the proposed technique. Simulation results, based on GlomoSim, indicate that this technique is very effective in improving network performance. It also works well in disciplining defecting hosts. More importantly, the success of the proposed technique does not rely on reputation exchange and is thus both scalable and robust.

REFERENCES

[1] B. Awerbuch, D. Holmer, C. Nita-Rotaru and H. Rubens. An On-Demand Secure Routing Protocol Resilient to Byzantine Failures. ACM Workshop on Wireless Security (WiSe) 2002.

[2] S. Buchegger and J. L. Boudec, IBM Research Report: The Selfish Node: Increasing Routing Security in Mobile Ad Hoc Networks. RR 3354, 2001.

[3] S. Buchegger and J. L. Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation of Nodes – Fairness In Dynamic Ad Hoc Networks. In Proceedings of

IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC), Lausanne, CH, June 2002.

[4] . S. Buchegger, H. L. Boudec, Coping with False Accusations in Misbehavior Reputation Systems for Mobile Ad-hoc Networks. EPFL Technical Report IC/2003/31.

[5] L. Buttyan and J. Hubaux. Enforcing Service Availability in Mobile Ad Hoc WANs. In Proceedings of IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC), Boston, MA, USA, August 2000.

[6] L. Buttyan and J. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. Technical Report DSC/2001/046, EPFL-DI-ICA, August 2001.

[7] Stephan Eidenbenz and Luzi Anderegg, Ad hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents. In Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom 2003), September 2003.

[8] Y. Hu, A. Perrig, and D. B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. Technical Report TR01-383, Department of Computer Science, Rice University, December 2001.

[9] Y. Hu, D. B. Johnson and A. Perrig. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless AdHoc Networks. In Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA 2002). IEEE, Calicoon, NY, June 2002.

[10] J. Hubaux, L. Buttyan, and S. Capkun. The Quest for Security in Mobile Ad Hoc Networks. In Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) 2001.

[11] D. Johnson and D. A. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In T. Imielinski and H. F. Korth, editors, Mobile Computing, pages 153--181. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1996.

[12] V. Kärpijoki, Security in Ad Hoc Networks. In Proceedings of the Helsinki University of Technology, Seminar on Network Security, 2000.

[13] S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In Proceedings of MOBICOM 2000, pages 255-265, 2000.

[14] P. Michiardi and R. Molva. CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad Hoc Networks. Sixth IFIP Conference on Security Communications and Multimedia (CMS 2002), Portoroz, Slovenia, 2002.

[15] P. Michiardi and R. Molva. Prevention of Denial of Service Attacks and Selfishness in Mobile Ad Hoc Networks. Research Report N° RR-02-63. January 2002.

[16] P. Michiardi, R. Molva. Simulation-based Analysis of Security Exposures in Mobile Ad Hoc Networks. European Wireless Conference, 2002.

[17] X. Zeng, R. Bagrodia, and M. Gerla. GloMoSim: A library for parallel simulation of large-scale wireless networks. Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS '98), May 26-29, in Banff, Alberta,Canada, 1998.

[18] Y. Zhang and W. Lee. Intrusion Detection in Wireless Ad Hoc Networks. In Proceedings of MOBICOM 2000, pages 275-283, 2000.

[19] L. Zhou and Z. Haas. Securing Ad Hoc Networks. In IEEE Network magazine, special issue on networking security, Vol. 13, No. 6, November/December, pages 24-30, 1999.

[20] Sheng Zhong, Jiang Chen, Yang Richard Yang, Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad-Hoc networks. IEEE INFOCOM 2003.

[21] ANSI/IEEE Standard 802.11, 1999 Edition. 1999. http://standards.ieee.org/catalog/olis/lanman.html.

[22] Information Technology Laboratory, National Institute of Standards and Technology. The Keyed-Hash Message Authentication Code (HMAC).