SECURE ROUTING IN AD HOC NETWORKS AND A RELATED INTRUSION DETECTION PROBLEM

Wensheng Zhang, R. Rao, Guohong Cao, and George Kesidis
Department of Computer Science & Engineering
The Pennsylvania State University
University Park, PA 16802
E-mail: {wezhang,gcao,rao,kesidis}@cse.psu.edu

ABSTRACT

The intrinsic nature of wireless ad hoc networks makes them vulnerable to various passive or active attacks. Thus, there is no guarantee that a routed communication path is free of malicious nodes that will not comply with the employed protocol and attempt to interfere the network operations. In this paper, we survey the problem of secure routing in ad hoc wireless networks, and discuss the related techniques of cryptographic key distribution. However, no matter how secure the routing protocol is, it is still possible that some nodes are comprimised and become malicious. The presence of comprimised nodes, especially in nodes that are communication bottlenecks, limit the effectiveness of the described secure routing protocols. We therefore consider the problem of intrusion detection for such nodes. The intrusion detection problem and some solutions are described in detail for a concrete queueing model of medium access. The extensions of the solutions to address the problem in more general scenarios are also discussed.

1. INTRODUCTION

Wireless ad hoc networking has been the focus of much recent research due to the potential applications in civilian and military environments including battlefield, disaster recovery, group conference, and wireless office. In ad hoc networks, mobile nodes communicate with each other using multi-hop wireless links. Due to lack of infrastructure support, each node in the network acts as a router, forwarding data packets for other nodes. Most of the previous research in ad hoc networks focuses on the development of dynamic routing protocols that can efficiently find routes between two communicating nodes. The intrinsic nature of wireless ad hoc networks makes them very vulnerable to attacks ranging from passive eavesdropping to active interfering. There is no guarantee that a routed communication path between two nodes will be free of malicious nodes that will, in some way, not comply with the employed protocol and attempt to interfere the network operation. Unlike wired networks where additional protection mechanisms can easily be deployed in routers and gateways, a malicious node could paralyze the entire wireless network by disseminating false routing information. Most existing routing protocols cannot cope with disruptions due to malicious behavior. For example, any node could claim that it is one hop away from a given destination node, causing all routes to that destination to pass through itself. Alternatively, a malicious node could corrupt any route request (reply) packet and cause data to be misrouted. Intentionally falsified route discovery messages would result in denial-of-service (DoS) attacks to severely degrade the network performance. Furthermore, traditional security techniques such as *certification authorities* (CA) [3], [21] are difficult to support in ad hoc networks. The consequent absence of authorization facilities impedes the usual practice of establishing a line of defense, separating nodes into trusted and non-trusted.

There are two sources of threats to routing protocols [26]. The first comes from external attackers who can replay old routing information or inject false routing information to partition the network or increase the network load. The second comes from the compromised nodes inside the network. Since the compromised nodes are able to generate valid signatures [3], [21] using their private keys, they are much harder to detect and can create severe damage. They might advertise incorrect routing information to other nodes, and such false routing information could result in messages from all nodes being fed to the compromised nodes. For external attacks, intrusion prevention measures, such as encryption and authentication, can be used to reduce intrusions. However, encryption and authentication cannot defend against compromised mobile nodes, which may carry the private keys. Also, integrity validation using redundant information, such as those being used in secure routing [12], also relies on the trustworthiness of other nodes, which could likewise be a weak link for sophisticated attacks. In this paper, we survey existing techniques that provide secure routing in ad hoc networks. Because of the inefficiency of existing techniques, we present an intrusion detection mechanism that provides a second line of defense.

The rest of the paper is organized as follows. Section 2 gives a survey of the existing techniques on providing secure routing. In Section 3, we present our intrusion detection technique. Section 4 concludes the paper.

2. SECURE ROUTING PROTOCOLS FOR AD HOC NETWORKS

In this section, we first present key management protocols which can be used to support public key infrastructure in ad hoc networks. The section concludes with a survey of existing secure routing protocols.

2.1. KEY MANAGEMENT PROTOCOLS

Most secure routing protocols are based on the public key infrastructure [21] because of its superiority in distributing keys and in achieving integrity and non-repudiation. A crucial problem of public-key distribution is how a node obtains the public key of another node in the presence of an active attacker [13]. The well-known approach to solve this problem is based on a trusted entity called a Certification Authority (CA) [3], [21]. The CA has public/private key pairs, with public keys known to every node, and signs certificates binding public keys to nodes. Using a single CA to provide certification services for the entire network may not work well in ad hoc networks due to mobility. Also, the CA may become a bottleneck and suffer from single point of failure. As a result, many researchers look into the problem of distributing the CA service. In [7], the authors proposed to let users issue certificates for each other based on their personal acquaintances. An algorithm called “shortcut hunter” was presented to construct local certificate repositories such that any pair of nodes can find certificate chains to each other in their merged repository with high probability even if the size of the local repositories is small compared to the total number of nodes in the system. Zhou and Haas [26] proposed a solution based on threshold cryptography [2], [19]. In their approach, the CA service is distributed over a certain number of nodes called servers. These servers collectively maintain the public/private key pairs for all mobile nodes. An $(n, t+1)$ threshold cryptography scheme (with $n \geq 3t+1$) allows n parties to share the ability to perform a cryptography operation (e.g., creating a digital signature) so that any $t+1$ parties can perform this operation jointly, whereas it is infeasible for at most t parties to do so even by collusion. In the solution provided by Kong *et al* [10], the CA's functionality is distributed among the local neighbors. A coalition of K (a system parameter) neighbors can serve as the CA and jointly provide certification service to a requesting mobile node.

2.2. SECURE ROUTING PROTOCOLS

Based on a trusted certificate authority, the authors of [18] proposed a solution to secure the routing protocol of ad hoc wireless networks. In their protocol, nodes get certificates from the CA to identify themselves to avoid spoofing and malicious route updates. To address the high overhead associated with obtaining and verifying the digital certificates, Hu *et al.* proposed a protocol [5] to secure on-demand routing protocols based on TESLA [16], an efficient broadcast authentication scheme that requires loose time synchronization. They also identified the wormhole attack [6], which may make most routing protocols unable to find routes longer than one or two hops. Based on the intuition that a receiver can determine if the packet has traversed a distance that is unrealistic with precise timestamp or location information, they provided a packet leash solution to solve the wormhole attack. The secure routing protocol (SRP) [15] does not rely on any certification authority or on a complete knowledge of keys of all network nodes. Based on a secure association between the source and the destination node, the protocol guarantees that fabricated, compromised, or replayed route replies would either be rejected or never reach the querying node.

Even with the secure routing protocols [5], [15], the source/destination may still set-up a route which goes through a misbehaving (malicious) node that agrees to forward packets but fails to do so. Note that such misbehaving nodes can behave normally during the route discovery phase but maliciously drop packets when the time comes to route data. In [12], the authors proposed to use a *watchdog* entity to identify misbehaving nodes and a *pathrater* mechanism to avoid routing packets through misbehaving nodes. When a node forwards a packet, the node's watchdog verifies that the next node in the path also forwards the packet. This can be accomplished by listening promiscuously to the next node's transmissions. If the next node does not forward the packet, it is misbehaving. The pathrater uses this knowledge of misbehaving nodes to choose the network path that is most likely to deliver packets. The solution in [23] exploits collaboration among local nodes to protect the network without completely trusting any individual node. In this approach, each node needs a token in order to participate in the network operations, and its local neighbors collaboratively monitor it to detect any misbehavior in routing or packet forwarding services. Upon expiration of the token, each node renews its token via its multiple neighbors based on its previous behavior.

Other researchers in this field also address the problem of intrusion detection [24], MAC layer misbehavior [11], and selfishness in packet forwarding [1].

3. INTRUSION DETECTION FOR AD HOC NETWORKS

Most previous work concentrates on using encryption and authentication to secure ad hoc networks. However, no matter how secure the protocol is, it is still possible that some nodes are compromised. In this case, intrusion detection can be used to provide another level of defense and it is an essential component to build highly secure wireless ad hoc networks. Hereafter, we study techniques to detect compromised nodes in ad hoc networks. These compromised nodes can be bottleneck nodes (denoted as X), which must be used as an intermediary for inter-group communication. That is, a bottleneck node is unavoidable from the perspective of routing packets between the two groups. We consider a situation where a bottleneck node has been hijacked and assume that a hijacked node, acting as a malicious intruder, does not wish to be detected. A malicious intruder may resort to dropping *data* packets to compromise communication in the network (data packets, of course, flow *after* routes have set-up). Thus, detecting (and then avoiding and removing) such intrusions has some intrinsic value. In this following, we will give a brief overview of the general intrusion detection problem and then focus on the scenario just described.

3.1. HISTORY OF INTRUSION DETECTION

Intrusion can be defined as any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource [4]. An *intrusion detection system* (IDS) can be categorized according to the data source [14] as: host-based, multihost-based and network-based. Based on the model of intrusion [20], an IDS can be classified as being either: an anomaly detection model in which the IDS detects intrusions by looking for activity that is different from a user's or system's normal behavior, or an misuse detection model in which the IDS detects intrusions by looking for activity that corresponds to known intrusion techniques (signatures) or system vulnerabilities.

Because of their intrinsic nature, wireless ad hoc networks are more vulnerable to an adversary's malicious attacks than wired networks [24]. These attacks range from passive eavesdropping to active interfering. Unlike in wired networks, attacks can come from all directions and target at any node and all layers. In [24], Zhang and Lee proposed a distributed and cooperative IDS architecture. In this architecture, all nodes participate in intrusion detection and response. Each node has an IDS agent. The IDS agents communicate with each other securely through a high-confidence channel. Each IDS agent runs independently and monitors local activities, gathering local audit traces and activity logs. It can detect intrusion locally and initiate responses if an intrusion is found. If an anomaly is detected or if the evidence is inconclusive, neighboring nodes can collectively participate in a global intrusion detection action. However, explicit criteria that can be used to detect malicious intruders and precisely how the neighboring nodes cooperate to detect a malicious intruder are not specified in [24].

3.2. THE NETWORK MODEL

The goal of the distributed protocol developed in this section is to detect malicious dropping of data packets by a bottleneck node. In [25], this problem of detecting malicious packet dropping was considered for the context of an intruder forwarding packets of TCP sessions in the wired Internet. The only rates of packet loss that are important in this context are those that are significantly higher than those due to buffer overflow because of the *actual* level of congestion at the bottleneck node. Because the assumption is that the intruder does not wish to be detected, he will continue to forward some packets traveling through communication paths in which it resides. The main difficulty is that the degree of congestion *at* the bottleneck node must be ascertained by its non-compromised neighboring nodes experiencing higher than expected packet loss. Clearly, the intruder would want to adopt a strategy that would make it difficult for other nodes to conclude that the packet loss is due to malicious nodes. To do this, the intruder may mimic packet dropping due to natural congestion. For this strategy to work, the intruder needs to assume that the other nodes may not be completely aware of the actual traffic load the intruder is experiencing.

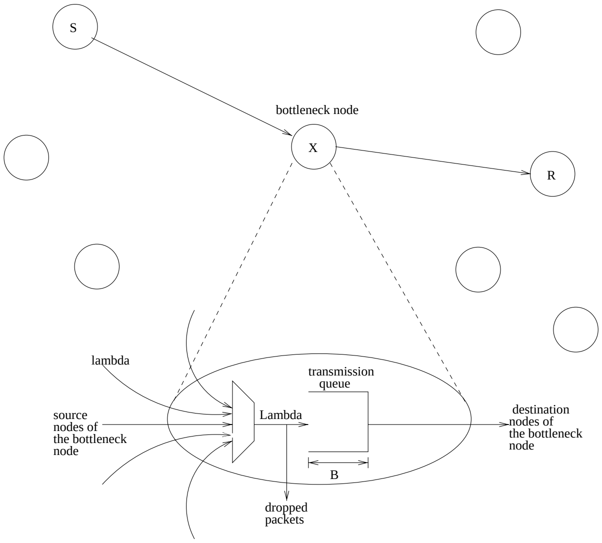
S
bottleneck node
X
R
lambda
transmission queue
source nodes of the bottleneck node
Lambda
destination nodes of the bottleneck node
B
dropped packets

Figure 1. Bottleneck node in a wireless ad hoc network

Consider a single unidirectional flow between two nodes,

S (sender) and R (receiver) out of direct communication range, see Figure 1. The (two-hop) path segment between these two nodes has a single intermediate node: the bottleneck node X under consideration. Consecutive packets of a flow are given consecutive sequence numbers that are encrypted using the private keys shared by S and R. In this way, R can be sure of how many packets S has actually sent and, therefore, how many packets have been lost in transit (dropped by X). Note that this flow could represent the aggregation of all flows that use a route containing the directed route-segment S, X, R. More specifically, let λ be the transmission rate in fixed-length packets per second of node S toward X constituting the flow. Let $r(i)$ be the index of the i^{th} successfully received packet by R from S for this flow. We assume that $r(1) = 1$. Node X may be handling several such active ($\lambda > 0$) flows giving a *total* arrival rate of packets to its queue of $\Lambda \geq \lambda$.

A basic assumption is that the nodes neighboring X are aware of its actual first-in-first-out (FIFO) buffer memory size (B packets) and mean service/transmission rate (μ packets per second). For stability of the its packet queue, $\Lambda < \mu$. We assume that the transmission patterns of each flow through X are a Poisson process which is consistent with an ALOHA-type (exponential backoff) medium access mechanism. We also assume that the queue is fully shared by all the flows through on X.

3.3. GROUNDS FOR SUSPICION AND VERIFICATION

Let T_i be the time that the i^{th} packet of the flow under consideration arrives at node R. At the receiving node R, the *sender's* mean transmission rate λ can be *estimated* by considering the arrival time of the flow's most recent packet, say $r(i)$, and the arrival time of the flow's first packet sequenced by 1

$$\widehat{\lambda} = \frac{r(i) - 1}{T_{r(i)} - T_1} \quad (1)$$

where we note here that the assumption that $r(1) = 1$ can easily be relaxed. Despite the fact that the destination may have received fewer than i packets by time $T_{r(i)}$ (i.e., that $r(i) > i$), this approach gives an unbiased estimator at the destination for the arrival rate λ for general stationary models of the network buffer. The reliability of this estimate clearly depends on the encryption of the packet sequence numbers. By taking stock of the authenticated sequence numbers of received packets, node R can immediately conclude whether a packet is missing if the received packet arrives out of sequence.

At time $T_{r(i)}$, the number of the flow's lost (not received) packets is $r(i) - i$, i.e., number of packets sent minus number received by time $T_{r(i)}$. Thus, the empirical probability of packet loss through the bottleneck node X is simply estimated by

$$\frac{r(i) - i}{r(i)}. \quad (2)$$

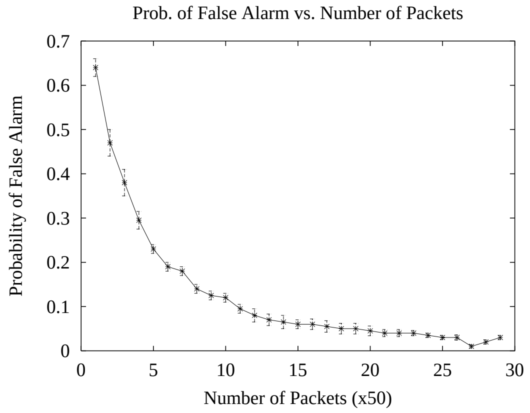

Figure 2. Bottle Node with a Single Flow (Traffic Intensity=0.9, Buffer size=7, and Confidence Interval=95%)

Assume that at node R, the current buffer memory capacity B and the queue's mean service rate μ of X are known. The current estimates of the arrival rate, $\widehat{\lambda}$, to X (from S) given in (1) and that of the total arrival rate, Λ, of all flows to X are also known at R. The loss rate for a Poisson source using a formula of Erlang and the rule that Poisson arrivals see time averages (PASTA) [22]:

$$\frac{(\Lambda/\mu)^B}{\sum_{k=0}^{B} (\Lambda/\mu)^k}. \quad (3)$$

So, we can *compare* (2) with (3) to check whether the bottleneck node is not maliciously dropping packets. If the empirical (observed) loss rate (2) is significantly greater (say by 50%) than that predicted by (3), then the receiving node suspects that the bottleneck node is a malicious intruder. That is, R suspects X if

$$\frac{r(i) - i}{r(i)} > \frac{(\Lambda/\mu)^B}{\sum_{k=0}^{B} (\Lambda/\mu)^k}(1 + \alpha) \quad (4)$$

for some choice of $\alpha \geq 0$, say $\alpha = 0.5$. Clearly, the probability of *missed detection* of a malicious intruder is an increasing function of α and the probability of a *false positive* (false alarm) is a decreasing function of α. Figure 2 shows the false positive probability verses number of received packets. Here, the bottleneck node is assumed to have a single queue per flow. The queue can hold at most $B = 8$ packets and has traffic intensity $\rho = \lambda/\mu = 0.9$.

3.4. THE VERIFICATION PROTOCOL

When node R measures an empirical rate of packet loss through the bottleneck node X exceeds a certain threshold value, say 5%, node R becomes suspicious. It will use the following protocol to ascertain/verify the true *total* traffic load (Λ) experienced by the bottleneck node X. First, node R sends node X a message requesting it to *broadcast* a special message to all of X's neighbors. This special message requests that all neighbors of X send to R a *secure* (encrypted) message communicating the traffic volume that they subject X to, i.e., to communicate the component rates that make up X's total load, Λ. That X actually forwards this message to its neighbors can be policed by R by simply monitoring X's transmissions immediately after its original request. In any case, it is in X's best interest to forward R's query, again assuming X wants to avoid detection (if it is, in fact, an intruder) or false accusation (if it is "innocent"). An authentication system based on private symmetric session keys can be used to prevent an intruder X from spoofing nodes, i.e., from sending R false traffic load information from fictitious neighboring nodes of X in response to R's query.

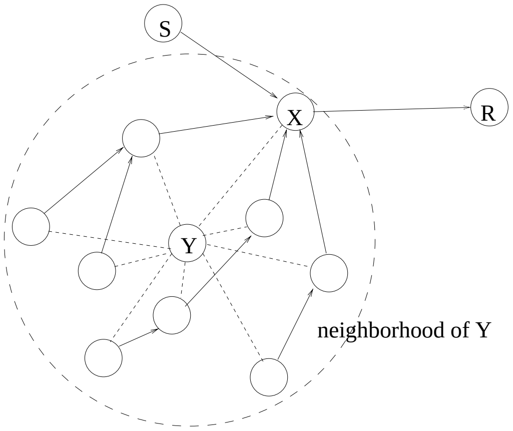

Figure 3. The bottleneck node collects traffic volume from node Y

It is possible that a node other than the communication bottleneck node, say Y, is the intruder. The intruder could then understate the volume of traffic it sends through the bottleneck node, in response to R's query, thereby increasing the likelihood that the bottleneck node will itself be (falsely) accused of being an intruder. This situation is avoided by asking each node to monitor the total traffic volume sent by each of its neighbors. Thus, X can collect the evidence about the actual volume of traffic from Y to itself and identify the true intruder, when being falsely accused by the victim. A collection process based on a tree is illustrated in Figure 3, and is explained as follows: Node X broadcasts a message $monitor(Y, X)$ to its neighbors. On receiving the message for the first time, the receiver joins the tree by setting the sender of the message as its parent node, and rebroadcasts the message. The message is propagated until the number of hops traveled by the message reaches a certain threshold. Then, each node on the tree reports to X, along the path in the tree, an encrypted message containing its monitored traffic volume from Y to X. To reduce the traffic, some aggregation work can be done during the collection process. After broadcasting the $monitor$ message, each non-leaf node in the tree sets a timer and waits for the data from its children. When the timer expires, it aggregates the collected data, merges them into one packet and sends the packet to its parent.

3.5. EXTENDED WORK

The previously described approach for detecting and identifying malicious intruders can be adapted for the case of a round-robin medium access mechanism (controlled by X) for which deterministic bounds on packet loss rates are available [8]. Instead of a fully shared queue, we could also consider a bottleneck node X employing a separate FIFO queue per active flow. Node X could use a kind of randomized scheduling [9] to multiplex the flows together for the purposes of transmission; in this case, for each flow, X would appear as a single $M/M/1/(B/N)$ queue with service rate μ/N where N is the *number* of active flows through X. Suspicious neighboring nodes would need to ascertain only the *number* of active flows (whether $\lambda > 0$) through X instead of the transmission rate (λ) of each flow. For a more practical wireless MAC setting, the effect of "model error" of (3) can be mitigated by a larger choice of α in the "suspicion rule" (4) at the expense of higher probabilities of missed detection. This trade-off is explored in [17] for various queueing models of the bottleneck node, X.

The previously described approach can also be extended to detect a malicious intermediate node, which is not a communication bottleneck node, on a path of three or more hops. To achieve this goal, two private keys are needed for each path, such that each node on a path shares a private key with the node two hops away from it on the path (Here, we assume that any two neighboring nodes on the same path can not be both malicious). Similar to the previous approach, each node on a path assigns consecutive sequence numbers to the packets it issues or forwards, and encrypts it with its private key. On receiving a packet, each node uses its private key to decrypt the sequence number assigned by the upstream node two-hop away, and records the number. In this way, each node on the path can estimate the empirical probability of packet loss through its direct upstream node (see Equation (2)). When the probability becomes larger than a certain threshold, the node becomes suspicious and uses a verification protocol similar to that previously described to check if the upstream node is malicious. Here, the major difference between this verification protocol and the previous one lies in that the upstream node should claim its service rate μ, and show that the rate is reasonable by collecting and showing the service rate of all its neighboring nodes (Of course, the claimed service rate should not be much different from those of its neighboring nodes).

3.6. DISCUSSIONS

There exist some problems which make it difficult to use the proposed approach to detect and identify the nodes maliciously dropping packets. In the following, we discuss some of them.

Worse than dropping a part of packets, a malicious node may drop all the packets of a flow. Suppose that S is the closest node to X and X maliciously uses only enough energy to forward S's packets so that S itself can hear them. S might therefore wrongly conclude that X is forwarding his packets in good faith. To detect this kind of misbehavior, other nodes in X's transmission range need to be vigilant and verify that X is indeed forwarding S's packets with sufficient power to reach the next intended hop (e.g., R). This implies that S has to transmit to X with enough power to reach other nodes nearby X.

Mobility of nodes make it more difficult to detect and identify malicious nodes. As described previously, when a node is suspected of maliciously dropping packets, it needs to collect the traffic volume emanated from its neighbors. However, some of these nodes may have moved away. To address this problem, the suspected node can appeal to the neighbors of these nodes to obtain the information needed (Note that, there are some nodes which are relatively stable). Another method to address this problem is to let a node query the traffic volume information from its neighbors whenever the probability for it to drop packets exceeds a certain threshold.

4. CONCLUSIONS

This paper surveyed the the problem of secure routing in ad hoc wireless networks and discussed the related techniques of cryptographic key distribution. The presence of malicious intruders, especially in nodes that are communication bottlenecks, limit the effectiveness of the described secure routing protocols. We therefore considered the problem of intrusion detection for such nodes. The intrusion detection problem and some solutions were described in detail for a concrete queueing model of medium access. In particular, a specific rule determining grounds for suspicion and verification was given, and a protocol for verifying whether a node is malicious was presented.

References

[1] L. Buttyan and J. Hubaux, "Stimulating Cooperation in self-Organizing Mobile Ad Hoc Networks," *ACM/Kluwer Mobile Networks and Applications (MONET)*, to appear.

[2] Y. Desmedt, "Threshold Cryptography," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 449–457, July/August 1994.

[3] M. Gasser, A. Goldstein, C. Kaufman, and B. Lampson, "The Digital Distributed Systems Security Architecture," *Proc. of the 12th National Computer Security Conference*, pp. 305–319, 1989.

[4] R. Heady, G. Luger, A. Maccabe, and M. Servilla, "The Architecture of a Network Level Intrusion Detection System," *Technical report, University of New Mexico, Department of Computer Science*, Aug. 1990.

[5] Y. Hu, A. Perrig, and D. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Wireless Ad Hoc Networks," *ACM Mobicom*, Sep. 2002.

[6] Y. Hu, A. Perrig, and D. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks," *IEEE Infocom*, April 2003.

[7] H. Hubaux, L. Buttyan, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks," *ACM MobiHoc*, pp. 146–155, 2001.

[8] G. Kesidis, "ATM Network Performance," *Kluwer Academic Publishers, Boston, MA, Second Edition*, 1999.

[9] G. Kesidis, P. Konstantopoulos, and M. Zazanis, "Sensitivity Analysis for Discrete-time Randomized Service Priority Queues," *IEEE CDC Proc.*, 1994.

[10] J. Kong, P. Zerfos, H. Luo, S. Lu, L. Zhang, "Providing Robust and Ubiquitous Security Support for Mobile Ad-hoc Networks," *IEEE 9th International Conference on Network Protocols*, 2001.

[11] P. Kyasanur and N. Vaidya, "Detection and Handling of MAC Layer Misbehavior in Wireless Networks," *Technical Report, UIUC*, Aug. 2002.

[12] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *ACM MobiCom*, Aug. 2000.

[13] A. Menezes, P. Oorschot, and S. Vanstone, "Handbook of Applied Cryptography," *CRC Press*, 1997.

[14] B. Mukherjee, L. Heberlein, and K. Levitt, "Network Intrusion Detection," *IEEE Network*, vol. 8, no. 3, pp. 26–41, May/June 1994.

[15] P. Papadimitratos and Z Haas, "Secure Routing for Mobile Ad Hoc Networks," *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, Jan. 2002.

[16] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," *IEEE Symposium on Security and Privacy*, May 2000.

[17] R. Rao and G. Kesidis, "Intrusion Detection Based on Packet Dropping Patterns in Multi-hop Wireless Networks," *Technical Report, EE Dept. PSU*, 2002.

[18] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, and E. Belding-Royer, "A Secure Routing Protocol for Ad Hoc Networks," *IEEE Int'l Conf. on Network Protocols (ICNP)*, Nov. 2002.

[19] A. Shamir, "How to Share a Secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[20] A. Sundaram, "An Introduction to Intrusion Detection," *ACM Crossroads Issue 2.4*, Apr. 1996.

[21] J. Tardo and K. Algappan, "SPX: Global Authentication Using Public Key Certificates," *IEEE Symposium on Security and Privacy*, pp. 232–244, May 1991.

[22] R. Wolff, "Stochastic Modeling and the Theory of Queues," *Prentice-Hall, Englewood Cliffs*, 1989.

[23] H. Yang, X. Meng, and S. Lu, "Self-Organized Network-Layer Security in Mobile Ad Hoc Networks," *ACM Workshop on Wireless Security*, Aug. 2002.

[24] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad-Hoc Networks," *ACM Mobicom*, pp. 275–283, Aug. 2000.

[25] X. Zhang, S. Wu, Z. Fu, and T. Wu, "Malicious Packet Dropping: How it Might Impact the TCP Performance and How We Can Detect It," *IEEE ICNP*, 2000.

[26] L. Zhou and Z. Haas, "Securing Ad Hoc Networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, November/December 1999.