Available online at www.sciencedirect.com
ScienceDirect
Ad Hoc Networks 6 (2008) 578–599
Ad Hoc Networks
www.elsevier.com/locate/adhoc

Threshold-based intrusion detection in ad hoc networks and secure AODV ☆

A. Patwardhan [a,*], J. Parker [a], M. Iorga [b], A. Joshi [a], T. Karygiannis [b], Y. Yesha [a]

[a] Computer Science and Electrical Engineering Department, UMBC, 1000 Hilltop Circle, Baltimore, MD 21250, United States
[b] Computer Security Division, National Institute of Standards and Technology, Gaithersburg, MD 20899, United States

Received 1 November 2006; received in revised form 28 February 2007; accepted 9 May 2007
Available online 18 May 2007

Abstract

Mobile ad hoc networks (MANETs) play an important role in connecting devices in pervasive environments. MANETs provide inexpensive and versatile communication, yet several challenges remain in addressing their security. So far, numerous schemes have been proposed for secure routing and intrusion detection, with only simulations to validate them; little work exists, in implementing such schemes on small handheld devices. In this paper, we present our approach of securing a MANET using a threshold-based intrusion detection system and a secure routing protocol. We present a proof-of-concept implementation of our IDS deployed on handheld devices and in a MANET testbed connected by a secure version of AODV over IPv6 – SecAODV. While the IDS helps detect attacks on data traffic, SecAODV incorporates security features of non-repudiation and authentication, without relying on the availability of a Certificate Authority (CA) or a Key Distribution Center (KDC). We present the design and implementation details of our system, the practical considerations involved, and how these mechanisms can be used to detect and thwart malicious attacks.

Keywords: MANETs; Secure routing; Intrusion detection; SecAODV

1. Introduction

Continued advances in hardware for miniature devices like Personal Digital Assistants (PDAs), mobile phones and converged devices like PDA-phones are adding significant computational, storage and communication capabilities to them. Parallel advances in both infrastructure based and ad hoc mobile networks are allowing these devices to interconnect, often spontaneously with other devices in their vicinity. These developments have led to efforts to provide services to these devices. Some of these use infrastructure based networks with client–server models [1,2]. Others have a broader (and longer term) vision of services in the vicinity being used in a peer-to-peer manner over ad hoc networks [3,4]. Application scenarios for the latter include providing connectivity to First Response Teams in

☆ This research was supported by NSF award 9875433, and grants from NIST and IBM.
* Corresponding author. Tel.: +1 410 4558680.
E-mail addresses: anand2@cs.umbc.edu (A. Patwardhan), jparke2@cs.umbc.edu (J. Parker), miorga@nist.gov (M. Iorga), joshi@cs.umbc.edu (A. Joshi), karygiannis@nist.gov (T. Karygiannis), yeyesha@cs.umbc.edu (Y. Yesha).

1570-8705/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2007.05.001

search and rescue missions in disaster affected areas, battlefields etc. Any infrastructure in such situations is non-existent, damaged, or compromised. Other scenarios involve ad hoc collaborations in busy public places such as airports and malls.

In most of these scenarios, the communicating devices may or may not have prior *trust associations* with each other. Due to the open and dynamic nature of such environments, identification and authentication become challenging problems. Devices can easily take "fake" identities and spoof IP addresses. Mobile devices can add a layer of security using cryptographic protocols for communication. However, identities of trusted devices are difficult to determine, since authentication services may not be available. Also even in the case of pre-distributed security credentials, a device can be captured or compromised and subsequently try to subvert the network of which it is a part.

In this paper, we present our implementation of a two pronged approach for protecting MANETs against attacks – secure the routing process and deploy IDSs on individual nodes throughout the network to detect misbehavior. This is a unique combination of known approaches viz. the concept of a *watchdog* [5] for intrusion detection in MANETS, and using security mechanism of decentralized cryptograhic generation of IPv6 addresses to enable unique and non-repudiable identities [6].

Until now, several intrusion detection schemes and routing protocols for MANETs have been proposed in literature, but their validation has been limited to simulations. Our main contribution is the actual implementation of an intrusion detection system (IDS) and an IPv6-based secure routing protocol SecAODV, that we have deployed and tested on Linux-based mobile handheld devices (iPAQs connected via 802.11 in ad hoc mode, see Section 6). This combination of a secure routing protocol and an IDS, complementing each other, to secure a MANET from most known attacks, is another significant departure from existing efforts described in literature, and to the best of our knowledge the first such implementation on actual devices/networks. We present a detailed analysis of issues involved in the implementation and deployment of SecAODV and IDS in our testbed. We also present interesting results that provide insights into practical considerations involved in deploying these on resource-constrained devices, that have not been addressed thus far, and are not apparent from simulations. The source code for the MANET IDS and SecAODV have been made publicly available under the UMBC Gnu Public License [7,8].

Our IDS scheme is based on an algorithm that employs a threshold-based anomaly detection scheme to detect intrusions [9]. We have extended the AODV protocol and added security features to it that help prevent most of the prevalent attacks on AODV. The IDS on the other hand monitors data traffic and helps detect malicious or chronically faulty nodes. This paper is an extension of our earlier work [10] and incorporates large scale simulation results. The simulation results are used to analyze the scalability of our approach and also bring out several important properties of MANET environments.

The paper is organized as follows. In Section 2, we discuss security threats to MANETs and the difficulties in dealing with them. We also discuss related work that addresses some of these problems in Section 3. We present the SecAODV protocol in Section 4. In Section 5 we present our threshold-based detection approach and describe in detail how anomaly detection and routing state monitoring can be used to provide secure routing and detect attacks on data traffic. We propose a distributed Intrusion Detection System (IDS) architecture for MANETs that will help detect and prevent attacks on data traffic. We present the prototype performance analysis in Section 6. In Section 7 we describe the concept of our snooping based IDS and simulation results. We describe our prototype implementation which demonstrates the viability of the our two pronged approach of a distributed IDS system and secure routing mechanism based on AODV called SecAODV. Finally, we conclude with lessons learned and ideas for future work.

2. Background

Our focus is on two goals: (i) ensuring a secure, reliable routing mechanism and (ii) protecting the reliability and fidelity of data transmissions along established routes.

2.1. Collaborative routing mechanisms

MANETs are comprised of mobile devices communicating via their wireless interfaces, and their topology is continuously changing. Several routing protocols have been proposed for such environments, that are resilient to these conditions and are efficient in establishing and maintaining routes. We

focus on *on-demand* or *reactive* routing protocols, which do not maintain any *a priori* network topology information. Typically, devices advertise their presence by broadcasting `HELLO` messages. Neighboring nodes discover each other by listening to these messages to update their set of directly reachable nodes. To be able to communicate with nodes beyond their own radio range, these nodes issue route requests. Route requests are processed by neighboring nodes and on-demand routes are created if the routing process succeeds. A combination of timeouts, frequency of usage, and reachability, in addition to protocol specific routing messages are used to update individual routing tables. Intermediate nodes have to cooperate in the routing process and also relay data once the routes have been setup. Essentially, each device functions as a router.

Numerous protocols of this type have been proposed and some have been implemented for IPv4. Among them are Dynamic Source Routing (DSR) [11], Temporarily Ordered Routing Algorithm (TORA) [12], Ad-Hoc On Demand Distance Vector (AODV) routing [13,14], Signal Stability-Based Adaptive (SSA) routing [15], to mention a few. However, the proposed routing protocols assume non-hostile environments, where nodes faithfully forward all packets, and malicious nodes are absent. We describe a first of its kind implementation of a secure routing protocol based on AODV that uses IPv6 called **SecAODV** described in Section 4.

2.2. MANET vulnerabilities and possible attacks

MANETs are inherently vulnerable to several kinds of attacks due to the open medium of communication, resource-constrained devices, and the collaborative nature of the routing process. MANETs basically function to provide connectivity to devices in the absence of infrastructure support using a shared open communication medium and rely on collaboration from participating devices in doing so. Devices participate in the MANET by complying to the specifications of the routing protocol. Lack of conventional mechanisms for identification and authentication for individual devices and reliance on unknown nodes for collaboration increases the vulnerabilities of the MANET connectivity and resources of the individual devices (like routing tables and message buffers).

Since MANETs have not yet been widely deployed, no actual data is currently available that allows comprehensive attack analysis. Huang and Lee [16] propose an attack analysis model for ad hoc networks that uses a taxonomy of anomalous events to detect and analyze attacks. Several possible attacks on MANETs have been identified in literature [17–19,16]. They can be broadly classified into two types: (i) *routing disruption attacks* and (ii) *resource consumption attacks*. A more detailed survey and discussion on current state of secure routing protocols has been presented by Hu and Perrig in [20].

Attacks can target various layers of the protocol stack. Resource consumption attacks that exploit vulnerabilities in the Medium Access Control (MAC) layer and Physical (PHY) layers to consume bandwidth and energy in order to starve resource-constrained devices, are examples of *sleep-deprivation attacks*. To prevent against such attacks, security mechanisms must be provided in the MAC and PHY layers; they cannot be repulsed at higher levels.

We focus only on attacks specific to the networking and application layers (routing process and data traffic). A detailed classification of the possible attacks can be found in [16].

2.3. Identities and key management

As computing becomes pervasive, people expect to access services and information anytime and anywhere. These systems lack centralized control, are not under any single administrative domain, and in addition their users are not all known *a priori*. Moreover, devices are only guaranteed to be able to communicate with peers in their vicinity – internet connectivity is 'limited.' In other words, as a device moves about and interacts with other devices, it cannot be assumed that the identity of all the devices it interacts with is known up front. Due to this inherent ad hoc connectivity in pervasive environments, it is not feasible to be dependent on conventional client–server methods of Identification and Authentication typically used in wired networks. It may not be possible to contact any trust/reputation authorities to validate identities during every encounter due to network partitions. Given the vast possible number of devices, it will also not be scalable to have all resources centrally registered and later authenticated, especially where no fixed networking infrastructure exists. Besides suffering from the inherent problems of centralized and federated architectures like lack of scalability and fault tolerance – total reliance on such external authorities to provide trust

related services will severely limit the use of the potentially abundant resources in other peers in a device's vicinity.

These constraints rule out maintaining security associations via ''web-of-trust'' in systems like PGP [21]. PGP provides a ''web-of-trust'' process by which a node can determine the validity and trust of public keys without actually confirming the authenticity of the key. For instance node A might trust the authenticity of a public key for node C based on the fact that node A verified (sometime in the past) the authenticity of node B's public keys, and node B has verified the authenticity of node C's public keys. There are public databases available on the Internet for determining these PGP authentication chains. While this is an elementary form of distributed trust, it still relies on connectivity to keying servers, which may not always be possible in pervasive systems. It also lacks mechanisms for immediate response to suspicious activity and revoking existing trust relationships. Partitions in ad hoc networks are common, and response to intrusions in the local neighborhoods must be local and immediate to ensure the correct and secure operation of the device.

To secure MANETs, unique and reliable credentials must be possessed by each node. Also, reliable authentication mechanisms must be incorporated to secure the routing process and to classify legitimate, malicious, and faulty nodes. Without reliable authentication, securing the routing process and intrusion response are not possible. To ensure correct and reliable operation of any MANET related activity, some mechanism must exist for non-repudiation and to verify authenticity of messages.

2.4. Intrusion detection challenges

Although encryption and signed headers are intrusion prevention measures, vulnerabilities remain nonetheless. An IDS further strengthens the defense of a MANET. A reliable IDS, operating within a MANET, requires that trust be established amongst collaborating nodes in the absence of any pre-existing trust associations.

MANETs present a number of unique problems for Intrusion Detection Systems (IDS). Differentiating between malicious network activity and spurious but typical problems associated with an ad hoc networking environment, is a challenging task. In an ad hoc network, malicious nodes may enter and leave the immediate radio transmission range at random intervals or may collude with other malicious nodes to disrupt network activity and avoid detection. Malicious nodes may behave maliciously only intermittently, further complicating their detection.

Traffic monitoring in wired networks is usually performed at switches, routers and gateways, but an ad hoc network does not have these types of network elements where the IDS can collect audit data for the entire network. A wired network under a single administrative domain allows for discovery, repair, response, and forensics of suspicious nodes. A MANET is most likely not under a single administrative domain, making it difficult to perform any kind of centralized management or control. Network traffic can be monitored on a wired network segment, but ad hoc nodes or sensors can only monitor network traffic within their observable radio transmission range.

Dynamic topologies make it difficult to obtain a global view of the network and any approximation can become quickly outdated. Mobility introduces additional difficulty in setting up a system of nodes cooperating in an IDS. A node's movements cannot be restricted in order to let the IDS cooperate or collect data; neither can a node be expected to monitor the same physical area for an extended period of time. A single node may be unable to obtain a large enough sample size of data to accurately diagnose other nodes. Fortunately, other nodes also gather data over time and create their own audit logs.

The loss or capture of unattended sensors and personal computing devices may allow for a malicious node to obtain legitimate credentials and launch more serious attacks. A node that sends out false routing information could be a compromised node, or merely a node that has a temporarily stale routing table due to volatile physical conditions.

Thus, reliable identification is an important necessity for any MANET IDS, since detected intrusions must be associated with specific entities. Otherwise, attackers can impersonate other legitimate nodes and the utility of the IDS becomes questionable.

3. Related work

3.1. Secure routing protocols

In practice, even a single malicious node can launch various kinds of attacks against its peers.

Attacks can span from single adversary, routing disruption attacks like *man-in-the-middle attacks*, to multiple, colluding adversaries, resource consumption attacks such as *denial-of-service attacks*. Therefore, MANETs are extremely vulnerable to attacks due to their dynamically changing topology, absence of conventional security infrastructures and open medium of communication, which, unlike their wired counterparts, cannot be secured. To address these concerns, several secure routing protocols have been proposed: SAODV [22], Ariadne [18], SEAD [23], CSER [24], SRP [25], SAAR [26], BSAR [27], and SBRP [28].

SAODV (Secure Ad-hoc On-demand Distance Vector) protocol [22] is an extension of AODV [13] that assumes the existence of a certified public key for each node. To protect routing messages against forgeries, all intermediate nodes cryptographically validate the digital signature appended to the routing message.

The Ariadne [18] protocol secures DSR by authenticating the sender of the message. It requires either a shared key among paired peers, a system-wide distributed public key for each node or a TESLA (Time-Efficient Stream Loss-tolerant Authentication) key for each node.

AODV and DSR are similar in most aspects of route discovery and maintenance, however differ in the way packets are routed. In AODV routing entries are maintained by nodes along the route from source to destination, whereas in DSR source routing is used, i.e. the routed packets contain the address of each device that the packet will traverse. It is noteworthy to mention two similar secure routing protocols based on DSR that have been proposed in literature viz., BSAR [27] and SBRP [28]. SBRP additionally requires using DNS servers to simplify network configuration. SecAODV, BSAR, and SBRP all use the IPv6 address auto-configuration feature and sign control messages to prevent tampering. Signed evidence is produced by the originator of the message and signature verification is performed by the destination, without any form of delegation of trust.

While BSAR has been simulated in ns-2, SBRP has no known simulation. Neither BSAR nor SBRP have any known implementations. SecAODV is based on AODV and routing decisions are made on a hop-by-hop basis. Consequently the IDS designed to monitor traffic does not assume knowledge of next hop of packets, unlike in the case of watchdogs for DSR.

Security features of SecAODV are based on the use of *Statistically Unique and Cryptographically Verifiable* (SUCV) identifiers proposed by Montenegro and Castellucia [6]. Each device produces its own Public and Private Key pair, and generates a statistically unique address by computing a secure hash over its own Public Key. Other nodes can verify the binding (ownership) of that cryptographic address with the corresponding Public Key by recomputing the same address from the provided Public Key of the node. We use SUCVs in the AODV routing process to protect against address spoofing and other kinds of routing disruption attacks. However, there is a limitation to the use of SUCVs – they merely provide a secure binding between the IPv6 address and the Public Key of the node. They serve as the unique identifier of a node in the MANET. However, nothing is really known about the node using that identity, i.e. there is no pre-existing enumeration of trusted SUCVs. We describe the implementation in detail in Section 4.

Various schemes have been proposed in literature to detect Sybil attacks including position verification, resource-consuming challenges, and position verification. Sybil attacks in general are difficult to detect without an identification authority or without making strong assumptions like resource parity of all devices [29]. However in specific kinds of ad hoc networks like Vehicular Ad Hoc Networks (VANETs) with predictable properties like restricted mobility it is possible to address Sybil attacks without making such strong assumptions. For example a combination of GPS location, SUCV-like identifiers, and position verification would be sufficient to detect Sybil nodes in VANETs. Measuring signal strengths or other similar distinguishing methods could help improve detection. These techniques however can be orthogonally implemented and incorporated into the intrusion detection process [29,30].

We assume that the nodes in the network are capable of verifying that unique identities belong to distinct participants.

3.2. Intrusion detection

Marti et al. [5] propose a "watchdog" mechanism that observes misbehaving nodes and a "pathrater" mechanism that helps routing protocols to avoid such nodes. However, it is also necessary to attribute misbehavior to particular entities for effective

response to intrusions. In our approach, the use of SUCVs ensures a secure binding between the IPv6 addresses and Public Keys of individual nodes.

Zhang and Lee [17] categorize host-based IDSs based on anomaly detection and misuse detection. Anomaly detection-based systems detect intrusions based on an established baseline of normal behavior. Misuse detection involves identifying attack signatures and usage patterns associated with known attacks. Also, communication patterns are different from wireline devices and mobile devices are often expected to operate in disconnected mode. Anomalies are not easily distinguishable from localized, incomplete, and possibly outdated information. So, prevalent anomaly detection schemes used in wired networks are not directly applicable in wireless ad hoc networks. Hence, they propose a new architecture for an IDS, based on IDS agents.

Several specification based IDSs have been proposed that model the behavior of the routing protocol using a Finite State Machine (FSM) and try to classify anomalies – deviations from expected behavior, as attacks. Tseng et al. [31] describe several attacks possible in the base AODV protocol. They illustrate the use of a finite state machine to detect anomalous behavior in order to determine attacks. They also suggest the use of an additional previous hop field to ascertain the source/path of AODV control messages.

Specification based IDSs model the behavior of individual nodes by monitoring all the inbound and outbound, data and control traffic, of the monitored node. These approaches require that models be built from the protocol specifications, to be able to classify intrusions. Also, it is assumed that the monitored node is continuously observable. Monitoring the correct routing state of another node by observing all inbound and outbound traffic is a complex process and will require significant resources. Additionally, practical difficulties like device mobility and/or radio interference are unlikely to allow continuous accurate monitoring of devices, further complicating the task of specification based IDSs.

Adversaries may faithfully abide by the SecAODV routing process yet the data traffic through the established routes is susceptible to being dropped or tampered with. It is necessary to detect such nodes that are chronically faulty or malicious. To protect against such attacks, we propose a *threshold-based* and *routing-protocol-independent* IDS.

4. Secure routing using SecAODV

4.1. Overview

SecAODV aims to resist a variety of attacks, including impersonation, replay, message-forging, and modification attacks. These issues are discussed in detail in Section 8.

SecAODV is a highly adaptive distributed algorithm designed for IPv6-based MANETs that does not require: (1) prior trust relations between pairs of nodes (e.g., a trusted third party or a distributed trust establishment), (2) time synchronization between nodes, or (3) prior shared keys or any other form of secure association. The protocol provides on-demand trust establishment among the nodes collaborating to detect malicious activities. A trust relationship is established based on a dynamic evaluation of the sender's "*secure IP*" and signed evidence, contained in the SecAODV header. This routing protocol enables the source and destination nodes to establish a secure communication channel based on the concept of *Statistically Unique and Cryptographically Verifiable* (SUCV) identifiers [6,27] which ensure a secure binding between IP addresses and keys, without requiring any trusted CA or KDC. The concept of SUCV is similar to that of Cryptographically Generated Address (CGAs) [32]. SUCVs associate a host's IPv6 address with its public key that provides verifiable proof of ownership of that IPv6 address to other nodes.

IPv6 was adopted for its large address space, portability and suitability in generating SUCVs. The address auto-configuration feature available in IPv6 that allows IP auto-configuration for the nodes on a need basis, is of special importance.

4.2. Working of SecAODV

The AODV protocol [13] is comprised of two basic mechanisms, viz., *route discovery* and *maintenance of local connectivity*. The SecAODV protocol adds security features to the basic AODV mechanisms of route discovery, setup, and maintenance. The route caching feature of AODV is however disabled for ensuring end-to-end verification (see 4.2.2).

Hu and Johnson [33] have shown that route caching provides significant performance benefits in terms of control message overhead (fewer RREQs and RREPs sent and received), faster route setups, etc. Cache timeouts and cache sizes also play a

significant role in improving the performance. Performance has also been shown to deteriorate drastically beyond certain cache timeouts. Providing strong security and high performance simultaneously has proven to be difficult, since optimization techniques like gratuitous RREP from known cached routes have to be disabled, since prior security associations with such intermediate nodes cannot be assumed.

4.2.1. Secure address auto-configuration and verification

To join a MANET, a node executes a script that sets its Service Set Identifier (SSID), then proceeds to install and configure all IPv6 and SecAODV related kernel modules, and finally starts the `aodvd` daemon. The daemon obtains its site and global subnet identifiers, and runtime parameters from a configuration file and/or from the command line. The `aodvd` daemon then generates a 1024-bit RSA key pair. Using the public key of this pair, the securely bound global and site-local IPv6 addresses are generated. To derive the addresses, a node generates a 64-bit pseudo-random value by applying a one-way, collision-resistant hash function to the newly generated, uncertified, RSA public key. However, only 62 bits out of the generated 64 bits are then used for the IPv6 address because 2 bits of the address space are reserved. The final IPv6 address is generated by concatenating the subnet identifier with the pseudo-random value derived from the public key and by setting the two reserved bits, according to RFC 3513 (2373) [34]. A source node uses the secure binding to authenticate its IPv6 address to an arbitrary destination. Upon completion of the RSA keys generation and IP address configuration, SecAODV can optionally broadcast `Hello`-type, signed messages to its neighbors to make its presence known.

The basic idea behind the secure binding between IPv6 addresses and the RSA keys is to use the 62 out of the 64 low order bits of the IPv6 address, which represent the host ID, to store a cryptographic hash of the public key assigned or generated by the device. The two bits of the interface identifier that are not set using the hash of the public key are: (i) bit 6 which is defined as the `universal/local` bit, and (ii) bit 7 which is defined as the `individual/group` bit. The bits numbering starts with 0 at the leftmost bit of the interface identifier [34]. The relayed message is then signed and the public key is then attached to it. The structure of the SecAODV message is presented in Fig. 1 and discussed in detailed below.

The SecAODV implementation follows Tuominen's design [35] which uses two kernel modules `ip6_queue`, `ip6_nf_aodv`, and a userspace daemon `aodvd`.

4.2.2. Secure route discovery and maintenance

A `Hello` message contains the node's sequence number, the lifetime, and the node's IP address. The RSA public keys (modulus and exponent) are `base-64` encoded in a tag-length-value format.

A source node must sign its sent packets with its private key and attach its public key to it for signature and IP address verification in order to further authenticate the contents of its packets to an arbitrary destination node.

A source node S that requests communication with another member of the MANET referred to as destination D – initiates the process by constructing and broadcasting a signed route request message `RREQ`. The format of the SecAODV `RREQ` message differs from the one proposed in [13], it additionally contains the RSA public key of the source node S and is digitally signed to ensure authenticity and integrity of the message (refer to Fig. 1). Upon receiving a `RREQ` message, each node authenticates the source S, by verifying the message integrity, and by verifying the signature against the provided public key. Upon successful verification, the node updates its routing table with S's address and the forwarding node's address. If the message is not addressed to it, it rebroadcasts the `RREQ`. When D receives the `RREQ`, it constructs a signed route reply message (`RREP`) addressed to the source node S, which includes the D's public key, as shown in Fig. 1. D then unicasts the `RREP` back to the neighboring node from which the `RREQ` was received. Upon receiving a `RREP`, any routing node verifies the destination D's IP address and signature against the included public key, updates its own routing table for D and routes it towards S. If a route entry for S does not exist or has expired, the message is dropped and an error message is sent back to the previous hop along the incoming path of the affected packet. If S does not receive any reply in a predetermined amount of time, it rebroadcasts new route requests. *Maintenance of local connectivity* mechanism is optionally achieved by periodically broadcasting `Hello` messages. As mentioned earlier, in our implementation these messages are

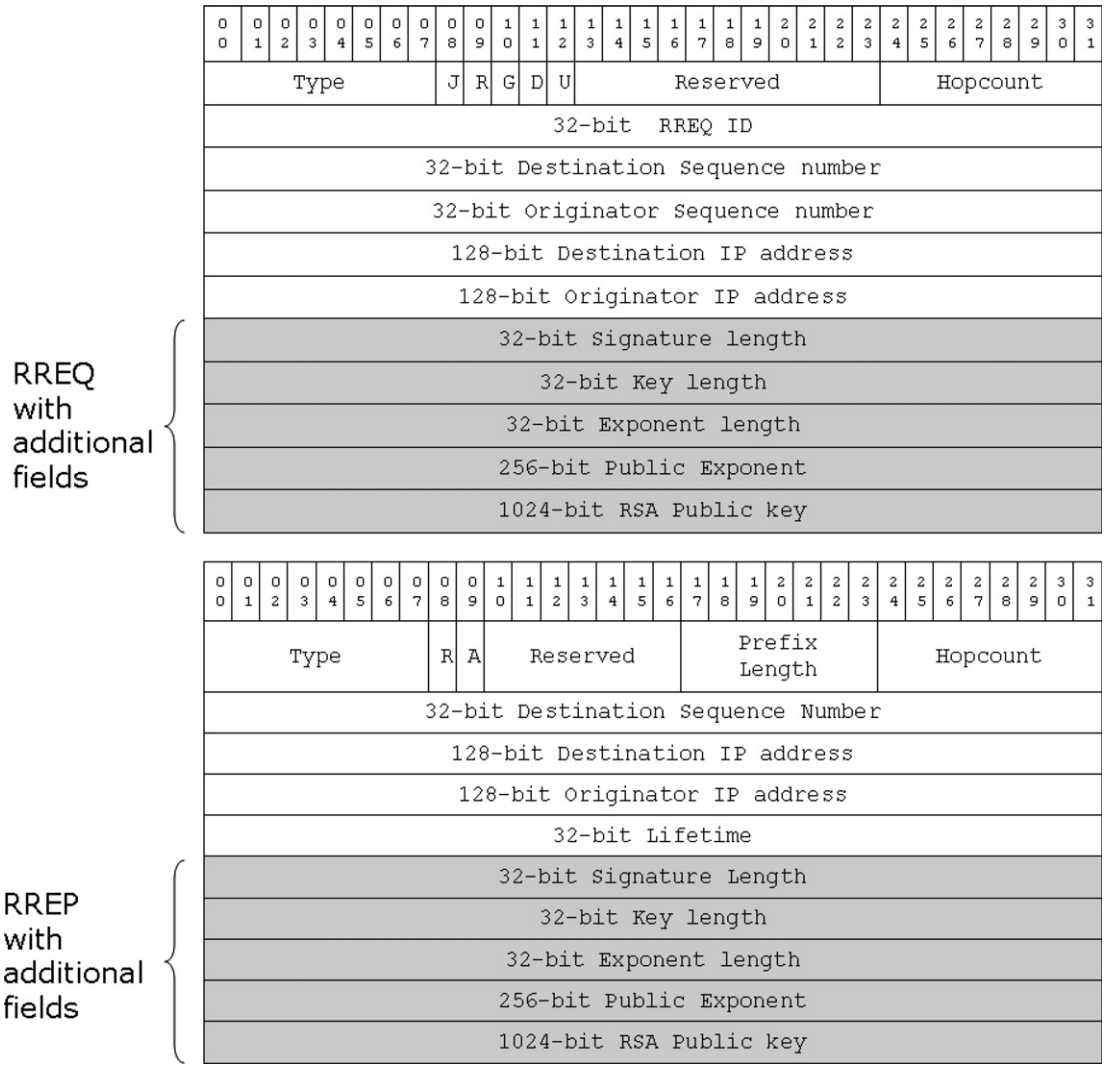

Fig. 1. SecAODV message formats.

signed and contain the sender's public key for authentication and message integrity verification.

For route requests (RREQ), the signed material contains: (i) the route request identifier, (ii) the destination sequence number, (iii) the originator sequence number, (iv) the destination IP, and (v) the originator IP. "Destination" is the address of the node for which the route is requested, and "originator" is the address of the node that requested the route.

For route replies (RREP), the signed material contains: (i) the destination sequence number, (ii) the lifetime, (iii) the destination IP, and (iv) the originator IP. "Destination" is the address of the node for which the route is supplied, and "originator" is the address of the node that requested the route.

In the AODV protocol if an intermediate node has a valid route to the destination in its own routing table, it can respond to that RREQ with its own RREP, which speeds up the setup time. However in order to prevent any man-in-the-middle attacks, in the SecAODV implementation we require that the RREP message be signed by the destination node.

When a message is received, the node (S, D or neighbor) first verifies that the IP address is derived from the presented public key, verifying in this way the binding between the IP and the public key. Secondly, if the IP address is validated successfully, the destination node verifies the packet's signature, thus verifying that the packet came from the listed IP address and not from an adversary that is maliciously masquerading as the listed IP address. In

the SecAODV implementation, the IP address and packet verification is implemented for all message types that might trigger routing table modifications (route requests, route replies, hello messages, route errors, etc.).

5. Intrusion detection in MANETs and prototype implementation

5.1. Assumptions and observations

We assume that interfaces have a promiscuous mode to monitor traffic of neighboring nodes. Key lengths are chosen to be sufficiently long, making it infeasible to compute or guess a private key knowing only the public key, but on the other hand do not make signature computation and verification computationally expensive for the mobile device. It is also assumed that normal packet drop rates can be dynamically determined and thresholds established to distinguish malicious behavior from trustworthy conduct. Thresholds can be set to drop rates that are low enough that impact is negligible, making such an attack ineffective. We do not, however, require MAC addresses to be unforgeable, since the SUCV identifiers provide secure bindings between IPv6 addresses and public keys. Identity is not determined by MAC addresses alone. Spoofing of IPv6 addresses and MAC addresses can be detected, since signature verification will fail unless private keys have been compromised.

5.2. Design considerations

5.2.1. Universal deployment

A MANET IDS should be able to function on any mobile device participating in the MANET, and not require additional special or superior capabilities as compared to its peers. The IDS must be universally deployable and should ideally be able to dynamically adapt to existing capabilities of a device to maximize its effectiveness and efficiency.

5.2.2. Scalable monitoring

The effectiveness of a MANET IDS will depend on its scalability. Snooping on all packet traffic is prohibitively expensive for most resource-constrained mobile devices, especially when number of nodes within radio-range increase. Dense networks or larger radio-ranges of new wireless technologies will have a large number of neighbor nodes.

5.2.3. Platform for a collaborative IDS

Attacks by colluding adversaries are far more complex and difficult to detect. They can only be detected by collaborative IDS schemes.

Individual nodes with IDS deployments can only monitor within their radio-range. It is necessary to aggregate such data to detect anomalies and malicious colluding activity in the network through peer interactions. The IDS should enable collection of local audit data. Detection of colluding adversaries is beyond the scope of this work, however aggregated information of observed (mis)behavior from a majority of good nodes will provide the necessary foundation for such a collaborative IDS.

In order to implement a truly robust IDS, there will be a need to aggregate data from multiple architectural layers. Alarms and thresholds placed at the network layer can report on the detection of routing misbehaviors such as observed incorrect packet forwarding. The MAC layer may alarm on nodes that send malicious CTS messages designed to deny other nodes network access. The Transport Layer may contain signatures for known attacks such as the SYN flood.

Delegating collaboration and trust issues to the application level, the IDS agent should enable collection of local audit data. The notion of trust is determined through an aggregation of information collected from observing multiple layers providing input for evaluation algorithms at the Application Layer. Collaboration not only comes from within the node, but can be shared between nodes as trust and reputation values that are interchanged between nodes throughout the network.

5.2.4. Enabling a protocol specific IDS

The IDS should allow monitoring of packet traffic for specific protocols. Specific protocols behave in a predictable pattern. Intrusion detection makes use of these patterns to spot abnormal behavior and in some instances specific signatures indicating malicious activity. Some protocols are more likely than others to be used with malicious intent. For example, in TCP a *SYN flood* can use up available ports on the target machine effectively denying service.

5.3. Scope of IDS

In our implementation approach, we focus on detecting intrusions based on anomalous behavior of neighboring nodes. Each node monitors particu-

lar traffic activity within its radio-range. An audit log of all locally detected intrusions is maintained as evidence of misbehavior. Intrusions are associated with pairs of IPv6 and corresponding MAC addresses. Local audit data can then be aggregated by some centralized/distributed algorithm, to detect ongoing attacks. Such collective analysis is however subject to *Trust* issues, since the problem of identification and authentication remains. Rather, in our current implementation, we focus only on the local detection and response part, to provide a foundation for such a collaborative IDS. By virtue of the SUCV identifiers, we can confidently identify the misbehaving nodes and associate intrusions with them.

5.3.1. Proposed approach

We detect intrusions by neighboring nodes by their deviation from known or expected behavior. When nodes act as forwarding nodes, offering routes to other destinations, it is expected that those node actually forward data packets, once a route through them is actually setup. Nodes are expected to retransmit the message without modifying the payload towards the intended recipient. We can categorize packet traffic into control packets that exchange routing information, and data packets. Depending on what routing protocol is being used, routing information may or may not be contained in the control packets, e.g., in DSR the routing information is present in the data packets; AODV on the other hand, does not have such information. Regardless of how routes are actually setup, data packets should not be modified, with the exception of some fields like hopcount in the IPv6 header. A node can thus monitor most of the packet traffic of its neighbors in promiscuous mode, while they are in radio-range. A node receiving packets but not forwarding them can be detected. We monitor AODV control messages and data stream packets only. We do not monitor control messages for faithful retransmissions. Since control messages are signed by the senders, modifications will be detected in the signature verification at the receiver.

5.3.2. Intrusion response

The purpose of intrusion detection is to isolate misbehaving nodes and deny them network resources. Nodes may be maliciously dropping packets or may have a genuine problem that prevents them from forwarding packets. Chronically faulty or malicious behavior, however, can be distinguished from transient failures by monitoring their activity over a period of time and setting thresholds. Such nodes are then deemed malicious and denied network resources. This can be done in two ways viz. unilaterally ignoring all traffic to or from a malicious node, and calling a vote on other members in the MANET to decide upon the eviction of a suspected node from the MANET [9]. Though this is a design goal, the collective response part has not yet been implemented.

5.4. Stateful packet monitoring

We use the packet capture library, `libpcap` [36–38], for capturing packets. As shown in Fig. 2a the captured raw packets are filtered to get only IPv6 using the protocol header field in the MAC header. Further filtering is used to separate AODV and TCP packets. We restrict ourselves to monitoring TCP data streams.

5.4.1. Discovering neighbors

The AODV control messages include special kind of `RREP` messages called `Hello` messages. These

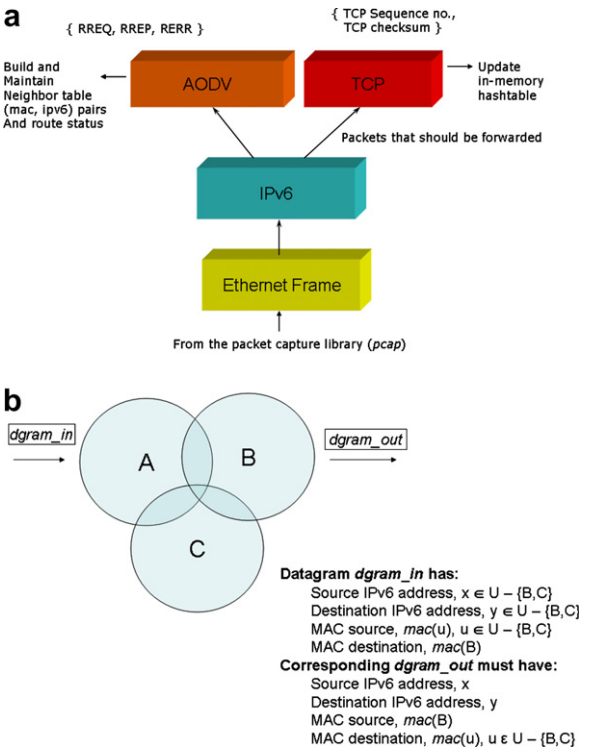

Fig. 2. Intrusion detection: (a) packet filtering and monitoring and (b) monitoring traffic in radio range.

messages are broadcast by the nodes at periodic intervals. Nodes can discover their neighbors using these messages. Also, if a neighbor moves away, the node will cease to receive it's neighbor's `Hello` messages and thus update its routing tables. We use these messages to build neighbor tables, which consist of tuples of the form `(MAC address, IPv6 address, drop_count, route_state)`, as shown in Fig. 2a. `(MAC address, IPv6 address)` constitute the unique key. This table is kept updated by monitoring `Hello` messages and `RERR` messages. More details on route maintenance and timeouts can be found in [13]. Data traffic of active neighbor nodes is monitored.

5.4.2. Monitoring data traffic

As shown in Fig. 2b we monitor data packets that need to be forwarded. Referring to Fig. 2b, consider nodes A, B and C within radio-range of each other. Without loss of generality, let C be the monitoring node, and B be the target of monitoring. A is sending a datagram via B to some other destination. B is acting as an intermediary node forwarding packets on behalf of A. Consider the datagram *dgram_in* sent by A to B. *dgram_in* will have MAC source address of A, MAC destination address of B. But the destination IPv6 address will not be that of B, since B is not the intended recipient of *dgram_in*. Now consider the datagram that B forwards after receiving *dgram_in*. *dgram_out* will have the MAC source address of B, however the source IPv6 address in the datagram will be that of A, and not B. In fact, *dgram_in* is a datagram that B is expected to forward and *dgram_out* will be that expected datagram sent out by B, onward to its intended recipient. Packets of specific protocols can be selectively monitored using the protocol field in the IPv6 header for filtering. C being the monitoring node, will first record *dgram_in* and watch for B to transmit *dgram_out*. The processing and queuing delay at B, may vary depending on congestion and CPU load on B. Under normal circumstances, B will transmit *dgram_out* within a reasonable amount of time. If B fails to do so, then C can infer that B must have dropped the packet. Another possibility is that B mangles the packet. When matching *dgram_in* and *dgram_out* for a particular protocol it is important to match all fields that should not be changed by B. If B maliciously mangles the packet, the original *dgram_in* will not match any *dgram_out*. C detects mangling by looking at the TCP sequence number, checksum and byte count.

5.5. Practical considerations

For the IDS to be effective it has to be scalable. A mobile device can get overwhelmed quickly if it starts monitoring all packets in its neighborhood in promiscuous mode. A large amount of data traffic in dense networks cannot be efficiently monitored by a resource-constrained mobile device. It may be possible in certain situations to have a list of suspects that can be watched instead of all the nodes in the neighborhood. Another possibility is to monitor a random choice of neighbor nodes. Alternatively, random packets can be watched to make the IDS scalable. Also the monitoring node needs to have efficient data-structures to monitor traffic efficiently in promiscuous mode. We also have to account for the buffering capacity of nodes. Our experiments showed that during periods of congestion, or route changes, a large number of packets get buffered by intermediate nodes. Buffered packets are those that a node will watch for to be retransmitted. The mobile device is constrained in how many packets it can watch for, so a timeout is associated with each packet being watched. On a timeout, the monitoring node deems such packets to be dropped. However, if these timeouts are too short, the IDS will yield a large number of false positives. We use thresholds to distinguish between intrusions and normal behavior. Thresholds can be used to account for temporary anomalous behavior due to congestion.

5.6. Threshold-based detection

Using threshold-based detection will potentially allow a malicious node to go unnoticed if it drops a few packets intermittently. However, the damage caused by such intermittent packet drops will be acceptable and will not significantly affect the MANET. If a node exceeds a small threshold of such allowed "misbehavior" it will be detected and classified as intrusive. An attacker cannot significantly disrupt communication while staying under the detection-thresholds, however will be detected if the threshold is crossed, i.e. the impact of such an attack will be negligible by choosing an appropriate threshold.

Thresholds allow for shorter timeouts, for packets being watched, since most packets are expected to be retransmitted immediately. In periods of congestion, a node may queue packets (to be retransmitted later) instead of transmitting them

immediately, causing the monitor to assume that the packets have been dropped; allowing a reasonable timeout period for retransmission reduces such cases. Also, each packet thus buffered on a neighbor node corresponds to the same packet being buffered by the monitoring node. In other words, each packet being watched accounts for memory consumed on the monitoring node, while the monitoring node waits for it to be retransmitted. A large number of neighbors buffering packets cause a significant aggregation of such packets on the monitoring node itself, which occupy memory until they are timed out. Not only will they result in false positives, they will also have occupied a large amount of memory on the monitoring node, and finally yielding possibly incorrect results. Deeming packets as "dropped," only after a timeout period, frees up memory in a reasonable amount of time, for monitoring newer packets, and reducing the overall memory requirements for monitoring, and at the same time minimizes false positives due to transient periods of congestion.

Consider the three relative movements of node C with respect to A and B, B being monitored, as shown in Fig. 3. The relative movement of the monitoring node with respect to its neighbors can cause false positives. In (i)–(iii) C is moving left horizontally monitoring B. When it gets out of range of B, it will continue to hear packets sent by A to B to be forwarded, but is out of range of B. Initially these will be registered as packets drops by B, however, the neighbor table will soon be updated since `Hello` messages from B will no longer be heard. The timeout periods are always chosen to be more than the `Hello` message intervals, thus accounting for such situations. In (a)–(c) the movement is towards B and away from A. So there will be no intrusions detected, since A will go out of range first. In (1)–(3) the movement is perpendicular and equidistant from A and B. Trivially, C can hear both A and B or none, so there cannot be any false positives.

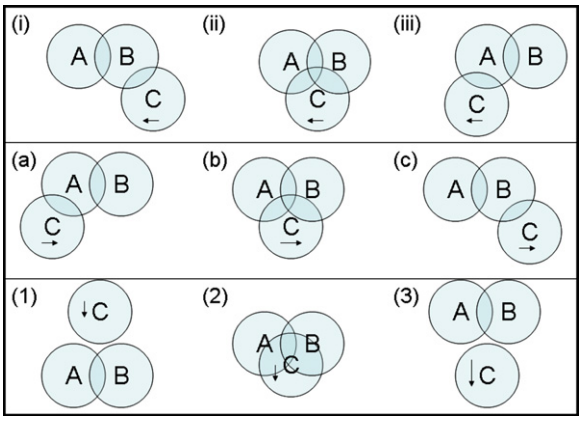

Fig. 3. Effects of mobility on IDS results.

5.7. IDS validation

To test the IDS functionality, we setup a node that could drop and/or mangle packets. This was done using the Linux kernel modules `ip6table_mangle` and `ip6_queue` (userspace packet queuing using `libipq`). Perlipq [39], a Perl extension to Linux iptables for userspace queuing via `libipq` was used. The process involves adding a rule to `ip6tables` to intercept all packets to be forwarded by the node, to be queued to userspace. Perlipq then allows these packets to be manipulated by the Perl program and then passed back to the kernel. The Perl program can mangle the payload, drop the packet or return it without modifying it.

Using the Perl program we configured the malicious node to have particular drop rates. The IDS immediately detected the dropped packets and reported them. If the drop rate exceeded the threshold value of the IDS, the IDS reported an intrusion and logged the incident. We observed that under normal traffic conditions hardly any packets are dropped by intermediate nodes when they are forwarding packets.

6. Prototype performance analysis

In our implementation, we used wireless cards that support the Prism2 chipset. We primarily used iPAQs in our testbed (specifications are provided in table 4(a)). We used the `ping6` utility for sending ICMP6 echo requests to determine reachability and response times. We setup the iPAQs in a linear chain using `ip6tables` to drop packets from specific MAC addresses at each node, to achieve this linear chain without physically separating the iPAQs out of radio range to get such a formation. The results of the ping tests are shown in Fig. 5. The AODV parameters used in the tests are shown in table 4(b).

Referring to Fig. 5, the response times of `ping6` packets are shown for destinations that are 1, 2 and 3 hops away. The first column labeled Basic AODV shows the response time of the AODV implementation that we used to build the secure version with

a

Handheld Device	iPAQ 3800 Series
Processor	206 MHz Intel StrongARM SA-1110 32-bit RISC Processor
Memory	64 MB SDRAM, 32 MB Flash ROM Memory
Wireless Access	Orinoco and Cisco Aironet 802.11b cards with wireless sleeves

b

Parameter	Value (ms)
NODE_TRAVERSAL_TIME	100
NET_TRAVERSAL_TIME	4000
NET_DIAMETER	20
PATH_DISCOVERY_TIME	2000
HelloInterval	2000
ActiveRouteTimeout	4000
DeletePeriod	20000
RouteTimeout	8000
ReverseRouteLife	8000

Fig. 4. Device details and parameters used in the testbed: (a) iPAQ specifications and (b) SecAODV parameters.

security features like signature verification turned off, but using the additional SecAODV header is shown. Finally the last column indicates the response time of SecAODV with all the security features enabled.

We note that the HUT AODV implementation [35] was tested in the AODV Interop Event [40] with only two hops. We got 100% packet loss with `ping6`, with more than two hops using HUT AODV. We fixed some of these problems, and used that as the basis for SecAODV. Column 1 in Fig. 5 enumerates the response times for the fixed version, sans the security features and column 2 enumerates those for SecAODV. In SecAODV, route caching is disabled for security reasons and thus suffers a performance setback compared to the non-secure version.

1 hop	Basic AODV	SecAODV
Min.	2.2s	2.2s
Max.	4.7s	19.7s
Avg.	2.76s	10.14s
2 hops	Basic AODV	SecAODV
Min.	79s	71.1s
Max.	169.8s	205.6s
Avg.	123.89s	145.8s
3 hops	Basic AODV	SecAODV
Min.	125.8s	122.4s
Max.	218.3s	269.5s
Avg.	168.67s	167.95s

Fig. 5. Ping6 response times in seconds using basic AODV version and SecAODV.

Comparing columns 1 and 2 in Fig. 5 we can observe that SecAODV does not significantly add to the routing overhead and/or cause packet loss as compared to the insecure version. We observed a large packet loss of ICMP6 packets in the original version. SecAODV however does not further add to the packet loss, the packet loss remained exactly the same, though the response times can be seen to be increasing slightly owing to the additional computations of encryption and decryption. With faster processors and larger memories the decryption and signature verification will be much faster.

The apparent improvement in the response times in SecAODV (see Fig. 5), in the case of 3 hops is an anomaly, which we attribute to the limitations of the measurement process and the instability of the prototype software. Nevertheless, these results show that the additional effort of signature verification process in SecAODV does not adversely affect the routing process even on handheld devices with severely constrained memory and computation power (see Fig. 4a).

The source code for SecAODV and the MANET IDS is available for download under the UMBC GNU Public License [7,8].

Fig. 6a shows the data rates for encryption and decryption data rates using different RSA keylengths. Fig. 6b shows key generation time for RSA keys.

7. Large scale IDS simulation

We used Glomosim 2.02 [41] to simulate a large scale deployment of a MANET with IDS nodes deployed in it. Node placement and travel was restricted to a 150 m × 150 m area. 802.11 was chosen

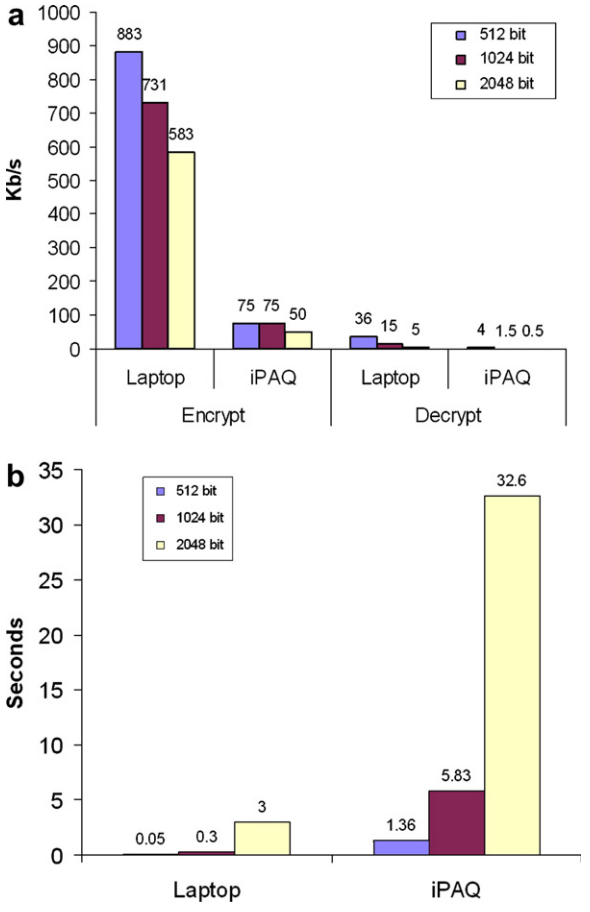

Fig. 6. RSA computations: (a) data rates for encryption and decryption using RSA keys and (b) RSA key generation time.

as the MAC layer protocol with each node having a range of approximately 30 m; no fading model was used. AODV was used as the routing protocol, and simulation times for each test was fixed to 300 s. Application traffic generated was the same for all tests. We followed the same traffic patterns used by Marti et al. [5], originally used by Broch et al. for performance comparisons of AODV and other routing protocols [42].

Briefly, the application traffic consisted of 10 constant bit rate (CBR) connections. Four of the nodes were sources of 2 CBR streams each, and two more with one CBR stream originating from them. Ten other nodes distinct from the sources served as the endpoints of those 10 CBR streams (a slight variation from [42] where there are only 9 receiver nodes, one of them with two CBR endpoints). The data rate for each connection in the simulation was 1 packet every 2 s, with a payload size of 512 bytes.

We then repeated the experiments with the same setup but replacing CBR by TCP flows.

We used the Random Waypoint Model for movement of the nodes, with a maximum speed of 5 m/s, minimum speed of 1 m/s, and maximum pause time of 15 s. Individual movement of nodes was specified in a trace file generated by BonnMotion 1.1 [43]. Use of the trace file allows for result correlation between tests, since nodes take identical paths for each test in which they participate. Complete parameters for the Glomosim simulation and the mobility trace files used for the scenarios are available online [44].

Maintaining the same initial starting positions for the existing nodes, the same mobility and traffic patterns, we studied the effect on neighbor table size, packets processed by IDS nodes, collisions, dropped packets, alarms generated, true positives, and false positives.

In each scenario, the same 25 IDS node observations are presented in the graphs below. The total number of nodes for the tests was varied from 100 to 300, with malicious nodes increasing from 10% to 50% of total nodes in increments of 10 percentage points. Bad nodes were configured to drop all data traffic yet participate correctly in the AODV routing process (grey holes). The bad nodes drop any traffic they are supposed to relay once included in the traffic path from sender to receiver.

Fig. 7a shows the mean neighbor table size as the total number of nodes is increased. Neighbor table size can be seen to grow from approximately 20 neighbors per node when total nodes are 100, to a neighbor table size of 100 when the number of nodes grows to 300. The growth in the neighbor tables is seen to be linearly proportional to the total number of nodes, as the density increases.

Fig. 7b shows the mean packet drop rates observed by the IDS nodes for 0 to 50 percent bad nodes, with increasing number of participating nodes. It can be observed from Fig. 7b that even the case with no bad nodes present, around 5 packets on average are observed as dropped regardless of an increase in the number of nodes. Such observed loss can be attributed to noise, transmission errors, congestion, other environmental conditions, and to the mobility of the devices. We use this as our base case and accordingly set our detection threshold for the simulations to 5 packets per sampling interval. Alarms are raised only if this threshold is exceeded. Post processing is done to classify the alarms into true positives (correctly

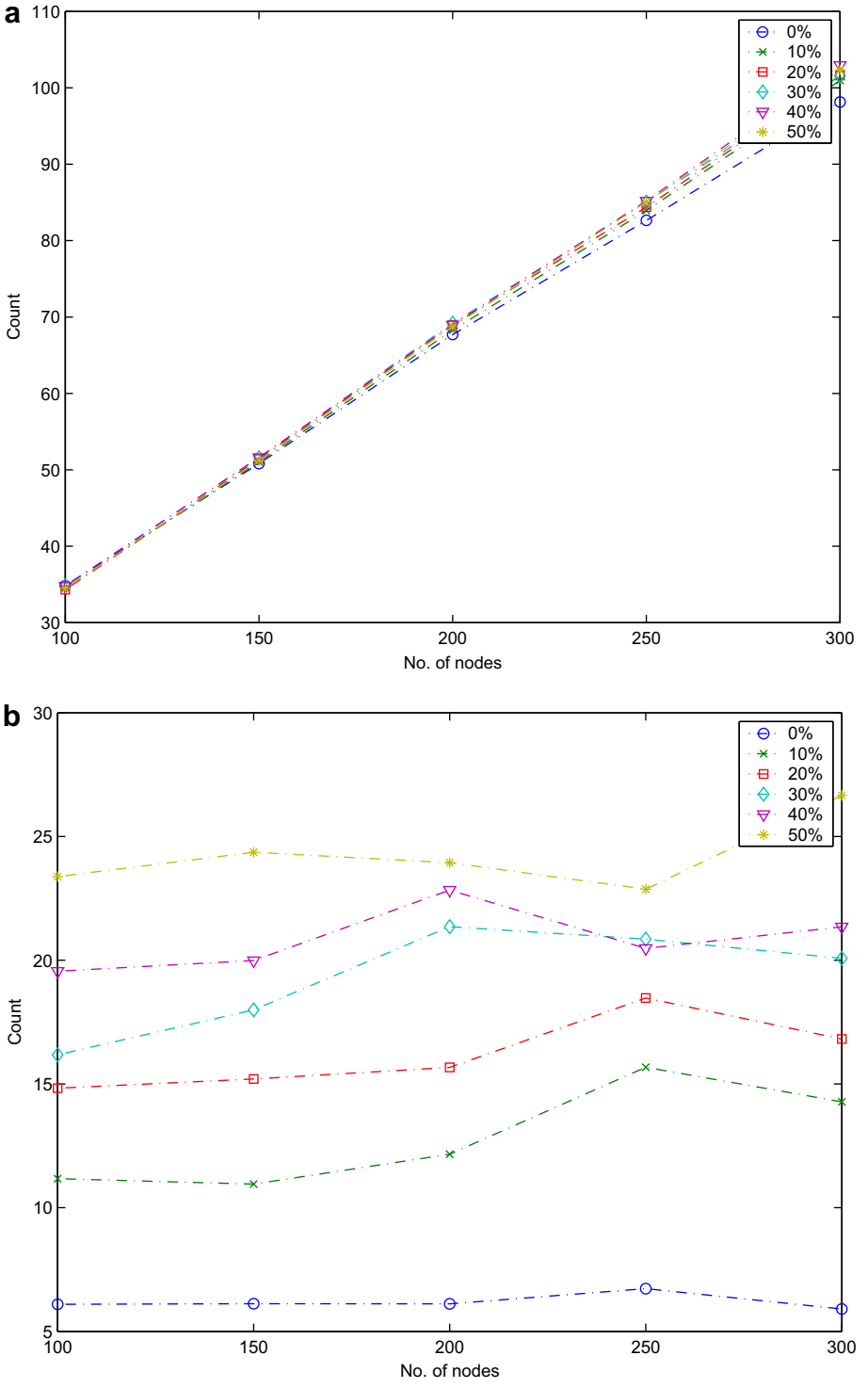

Fig. 7. Growth of neighbor tables sizes and dropped packets observed: (a) mean neighbor table size and (b) mean drops observed by IDS nodes.

identified malicious drops), and false positives (packets dropped for other reasons). Dropped packet data from the bad nodes is used for the classification.

Fig. 8a and b show the mean packet processing effort over the 25 IDS nodes for CBR and TCP traffic respectively. It can be observed that the number of packets processed decreases as the percentage of bad nodes (grey holes) is increased. With more nodes dropping data packets overall throughput decreases reducing the effort for the IDS nodes in detecting intrusions. Thus detecting grey holes requires much less effort. However in the case of nodes mangling (modifying) data packets, there will be no decrease in effort per IDS node, since all packets are retransmitted with possible modifications. In our current simulation bad nodes act as grey holes.

From Fig. 7a, mean neighbor table size for 100 nodes is around 20, and increases to up to 100 for 300 total nodes. With light traffic conditions like those used in the CBR simulations (Fig. 8a), the mean number of packets required to be processed by an IDS node per sampling interval (10 s)is seen to be fairly low, and decrease with increase in number of bad nodes, which is a manageable number even for resource-constrained devices. Possible techniques for handling situations when high throughput overwhelms devices have been discussed in Section 5.5.

Increasing the density of nodes around an IDS node is not seen to significantly affect the amount of processing required by the IDS, since the only additional packets processed are related to the routing protocol. The simulation results also shows a very low ratio of false positives with a static threshold of five dropped packets for intrusions and sampling period of 10 s used in the simulations. We believe this can be improved further by dynamically adapting to traffic conditions.

Regular nodes may occasionally be affected by adverse environmental conditions like high noise, high load, etc. leading to some packets being dropped or discarded. Malicious or chronically faulty nodes however exhibit continuous packet drops over a period of time. Thresholds, dynamically determined by monitoring current network conditions, would help account for such packet drops by regular nodes under transient adverse environmental conditions. This lower bound of acceptable packet loss, is chosen merely to allow for transient failures (packet loss) and minimize false positives. Consequently, for a specific deployment such a threshold should be chosen based on a variety of factors including network density, radio range, mobility speed and further modified by current traffic loads and environmental conditions like noise.

A larger sampling period would require more storage per (observable) traffic stream, and smaller one would require less storage – yet increases the possibility of false alarms. The IDS intrusion threshold can also be adapted to account for packets dropped due to congestion to reduce the number of false positives.

Further, from Fig. 8b, in which TCP flows are being monitored under identical conditions as Fig. 8a, the processing overhead is seen to be similar. In general, the IDS is not affected by the kind of payload contained inside the IP packets, but the focus here was on studying the effect of TCP and CBR traffic streams as the node density increases. It can be seen in case of both TCP and CBR streams that monitoring overhead actually decreases as the number of bad nodes increase.

Fig. 9a shows the packets observed by one of the 25 IDS nodes with 50 bad nodes and a total number of 100 nodes. The graph shows the number of packets processed, collisions, neighbor table size and dropped packets (observed) per 10 s time interval for the duration of the simulation. Packets processed are the total number of relevant packets the IDS was able to snoop on per time interval. Fig. 9b shows data plotted for the same node with dropped packets (observed), true positives, and false positives. The true positives indicate the number of bad nodes detected during the time interval, which in this case is at most one per time interval. Two true positives in the same time interval would indicate that two separate bad nodes were detected. Overall, the graph shows 10 true positives and one false positive. With static thresholds, false positives are likely to increase with increase in data traffic and congestion. False positives can be reduced by sensing congestion, collisions, and throughput in observable radio range – and adapting the IDS threshold.

Our simulation results verify that our prototype implementation is capable of monitoring traffic in realistic situations even under high node densities, and we can conclude from the simulation results that our prototype is scalable for a reasonably sized ad hoc network (hundreds of nodes) with light traffic conditions. Scalability of individual IDS nodes can be further increased and false positives minimized – by dynamically adapting the sampling interval and the intrusion thresholds to changes in traffic

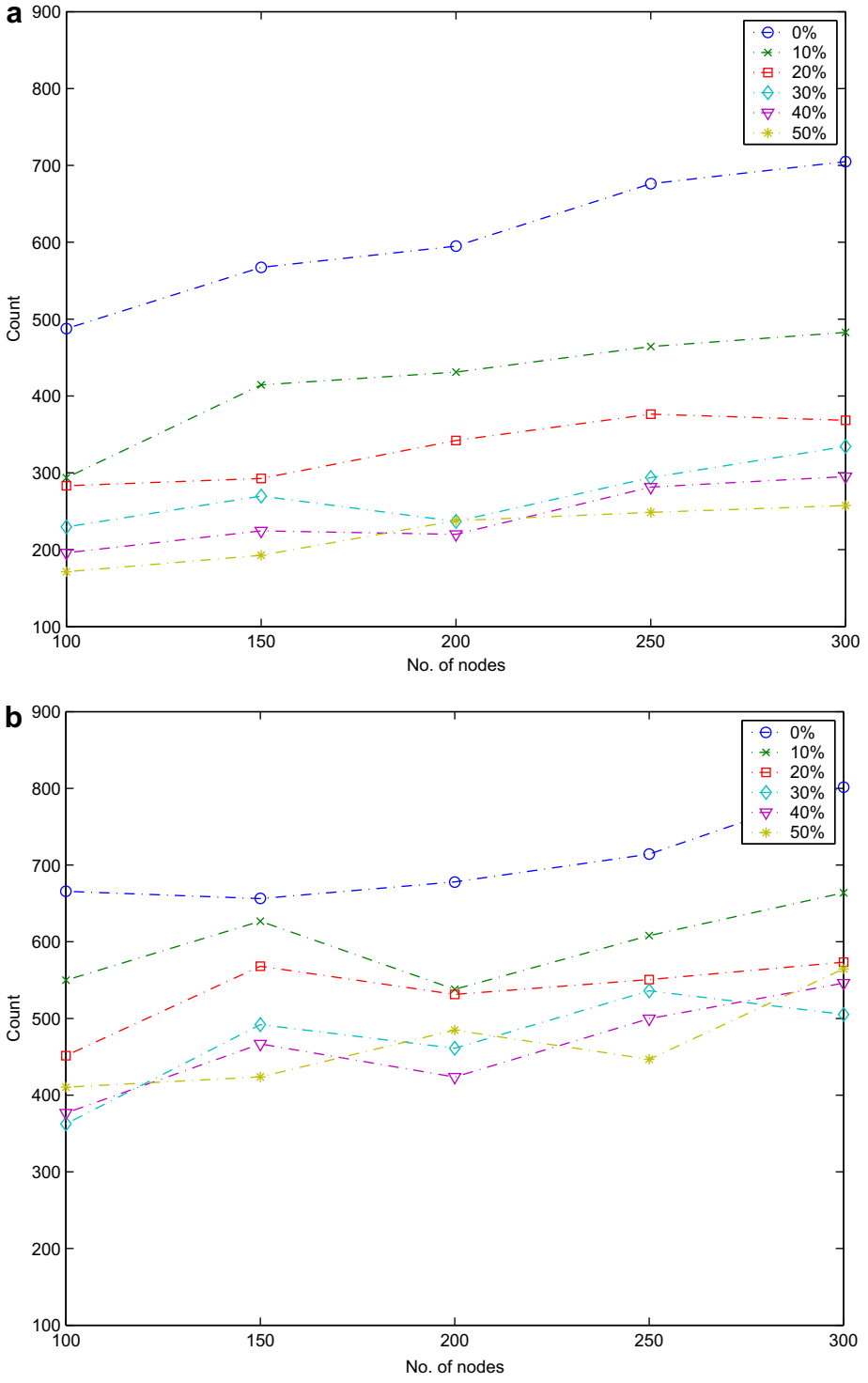

Fig. 8. Intrusion detection simulation: (a) packets processed in CBR traffic and (b) packets processed in TCP traffic.

conditions. Optimal sampling intervals can be adapted to the memory constraints of the node on which the IDS is deployed, and thresholds selected to minimize false positives.

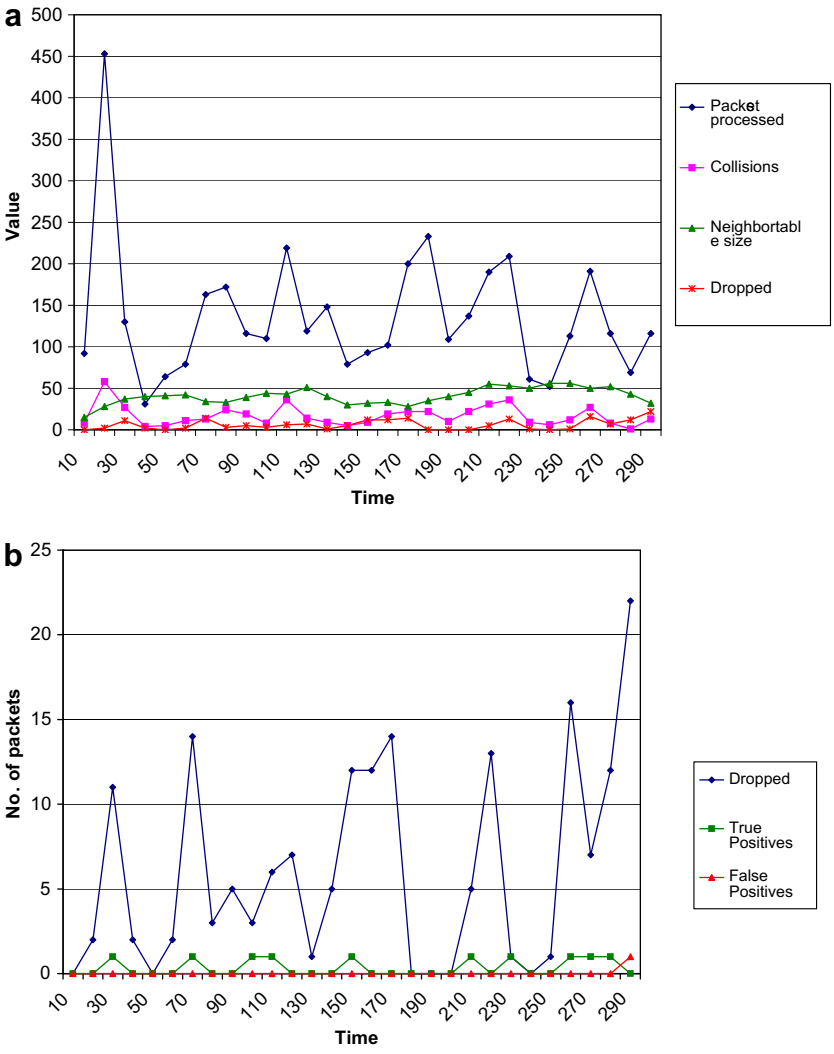

Fig. 9. Sample data from single IDS node: (a) IDS effort and (b) alarms generated.

8. Security analysis

8.1. SecAODV security analysis

In this section we discuss how SecAODV resists attacks by non-colluding adversaries. *Routing disruption attacks* in which an adversary attempts to forge a route request or a route reply by masquerading as another sender node or destination node are prevented, since either the IPv6 address verification or signature verification will fail. As long as the IPv6 address of a node and its public key are cryptographically bound, the attacker cannot successfully spoof another node's address unless the victim's private key is compromised.

An attacker might also try to initiate route replies without receiving a route request. This kind of attack has minimal impact since the attacked node can ignore packets from a node to which it did not request a route. Alternatively, an attacker can replay a cached route reply. This kind of attack is prevented since the protocol maintains status via sequence numbers contained in the signed header. AODV trivially eliminates duplicates and older messages using sequence numbers in the routing messages. Moreover, by including the destination and originator sequence numbers in the signed material, the SecAODV prevents *rushing attacks* [19] in which a malicious node rushes spurious messages in which the attacker modified any of these two fields making

the legitimate packet look old or as a duplicate. As long as the private keys of the end nodes are not compromised, the attacker is not capable of modifying any of these fields and thus immune to rushing attacks.

One kind of *resource consumption attack* is to initiate a lot of route requests, thereby causing congestion in the network. This attack can be mitigated by setting an "acceptance rate," thus limiting the number of route requests a node can accept and process per clock tick.

SecAODV also prevents *man-in-the-middle attack* by enforcing IP and signature verification. Unless the malicious node possesses the private keys of both end nodes, the attacker cannot launch a *man-in-the-middle* attack.

Routing disruption attacks in which an adversary attempts to forge a route request or a route reply by masquerading a source or destination node, are prevented since either the IPv6 address verification or signature verification will fail. As long as the IPv6 address of a node and its public key are cryptographically bound, an attacker cannot successfully spoof another node's address unless the victim's private key has been compromised.

Amongst the attacks on AODV, enumerated in [16], SecAODV protects against address spoofing, fabrication, and modification of routing messages, since signature verification is required before any routing message is used to update the routing tables.

8.2. IDS security analysis

While the use of signed control messages in a routing protocol like SecAODV can prevent routing disruption attacks, it is possible for an attacker to selectively drop only data packets (called *grey-hole attacks*). The IDS reinforces MANET security by detecting such kind of attacks. The IDS is able to detect dropped and mangled packets. Every time a packet is faithfully retransmitted the corresponding packet is removed from the *watch-list* by the IDS. Mangled packets will not match any packets the IDS is watching for retransmission, and thus timeouts will cause the IDS to deem those to have been dropped. In case of TCP streams, it is possible to distinguish mangled packets from dropped packets, using the TCP sequence number and byte count. From the sequence number in the TCP packet, we can determine which part of the stream the packet belongs to and use it to determine if the intermediate node has mangled the data in any way, or if the checksum is bad. It is important to establish thresholds for classifying detected intrusive behavior.

The IDS thus is able to monitor faithful retransmission of the packet payload, whether it is encrypted or not. The IDS however will not be able to monitor packets where link-level encryption is used at each hop, i.e. when the headers themselves are encrypted, without knowledge of link-level keys of all participating nodes.

A malicious node may change its own MAC address and IPv6 address periodically to evade detection. Thus, to go undetected, the attacker will need to change his/her IPv6 address very often, and incur the additional expense of computing a SUCV identifier every time. Consequently such an attack is largely ineffective, and quite expensive for the attacker. To maintain a sufficiently disruptive attack, the attacker would also have to follow the target node to remain along some crucial route of the targeted node, which is quite expensive for the attacker. Further such an attack essentially amounts to a Sybil attack [29] which we have assumed (see Section 5.1) can be detected by an additional mechanism [30]. The use of SUCVs makes impersonation very difficult, and rekeying SUCVs in the above circumstances would render such an attacks into a low level nuisance at the most.

Collaborative IDSs will perform best in a densely populated MANET with limited mobility, and will perform worse in a sparsely populated MANET with significant mobility. The effectiveness of a collaborative IDS depends on the amount of data that can be collected by each node. The longer the nodes are members of the MANET, the greater the availability of meaningful data for further analysis. The degree of mobility of each node in the network will also have a significant impact on the IDS's effectiveness. In a MANET with a high degree of mobility, if the number of routing error messages caused by legitimate reasons far exceeds the number of routing error messages caused due to the presence of malicious nodes, the effectiveness or benefit of such an IDS may be minimal. The damage that could be caused by a malicious node in highly mobile environment would, however, also be minimal since malicious routing messages would likely make up a small percentage of routing error messages.

The effectiveness of a collaborative IDS depends on the amount of data that can be collected individ-

ually. Longer presence increases the availability of meaningful data. However, the degree of mobility has a significant impact on the effectiveness of the IDS. Routing errors and packet drops due to increased mobility may mask malicious behavior, however malicious nodes cannot significantly affect routing either.

9. Summary and future work

We described the inherent vulnerabilities of mobile devices in MANETs and several attacks possible on such devices. We presented related work in this area and presented the design and implementation of our secure routing protocol SecAODV and IDS. The IDS is routing protocol independent, though in this case we have used SecAODV for routing. The role of the routing protocols is just to create and maintain routes. Even after protecting the network from routing disruption attacks, packet mangling attacks and grey holes, denial-of-service attacks that use MAC vulnerabilities to disrupt communication are still possible. However such attacks cannot be prevented at higher networking layers, rather security mechanisms need to provided in the MAC protocol itself.

Nodes can operate on their own, however for propagating information on misbehaving nodes a platform to enable collaboration for dissemination of such IDS data is needed. The scope of a host-based IDS deployed on a mobile device is limited to its radio range. We are currently working on implementing a collaborative IDS which will offer a collective response to misbehaving or intrusive nodes. In addition to using thresholds, we are also working on using signal strengths of neighboring nodes for detecting misbehaving nodes. Potentially an IDS may assume that a neighboring node is dropping packets, when in fact, the node simply moved out of range of the monitoring node. A low signal strength will help determine the distance of the neighboring node and thus help decide if a node is misbehaving or has simply moved out of range. Also it will be helpful in selection of nodes to monitor and increase the scalability and detection accuracy of the IDS. We are also looking into the additional energy consumption requirements when using secure routing protocols such as SecAODV and the effect of larger header sizes of routing messages (control overhead) on throughput and response times and ways to improve them.

References

[1] AT&T Wireless: Find People Nearby, February 2005. <http://www.attwireless.com/personal/features/organization/findfriends.jhtml>.

[2] Microsoft Mappoint, February 2005. <http://www.microsoft.com/mappoint/default.mspx>.

[3] F. Perich, A. Joshi, T. Finin, Y. Yesha, On data management in pervasive computing environments, IEEE Transactions on Knowledge and Data Engineering 16 (5) (2004) 621–634.

[4] D. Chakraborty, A. Joshi, Y. Yesha, T. Finin, Toward distributed service discovery in pervasive computing environments, IEEE Transactions on Mobile Computing 5 (2) (2006) 97–112.

[5] S. Marti, T.J. Giuli, K. Lai, M. Baker, Mitigating routing misbehavior in mobile ad hoc networks, in: Proceedings of MOBICOM, 2000, pp. 255–265.

[6] G. Montenegro, C. Castelluccia, Statistically Unique and Cryptographically Verifiable(SUCV) identifiers and addresses, 2002. <citeseer.ist.psu.edu/montenegro02statistically.html>.

[7] Source Code for MANET IDS. <http://ebiquity.umbc.edu/resource/html/id/104/Source-code-for-MANET-IDS>.

[8] Source Code for SecAODV. <http://ebiquity.umbc.edu/resource/html/id/105/Source-code-for-SecAODV>.

[9] J. Parker, J.L. Undercoffer, J. Pinkston, A. Joshi, On intrusion detection in mobile ad hoc networks, in: 23rd IEEE International Performance Computing and Communications Conference – Workshop on Information Assurance, IEEE, 2004.

[10] A. Patwardhan, J. Parker, A. Joshi, M. Iorga, T. Karygiannis, Secure routing and intrusion detection in ad hoc networks, in: Proceedings of the 3rd International Conference on Pervasive Computing and Communications, IEEE, Kauai Island, Hawaii, 2005.

[11] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: Korth Imielinski (Ed.), Mobile Computing, vol. 353, Kluwer Academic Publishers, 1996. <citeseer.ist.psu.edu/johnson96dynamic.html>.

[12] V.D. Park, M. S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: INFOCOM, vol. 3, 1997, pp. 1405–1413. URL <citeseer.ist.psu.edu/park97highly.html>.

[13] C. Perkins, E. Belding-Royer, S. Das, RFC 3561: Ad hoc On-Demand Distance Vector (AODV) Routing, July 2003. URL <http://www.ietf.org/rfc/rfc3561.txt>.

[14] L. Klein-Berndt, A Quick Guide to AODV Routing.

[15] R. Dube, C. Rais, K. Wang, S. Tripathi, Signal stability based adaptive routing (ssa) for ad hoc mobile networks, February 1997. URL <citeseer.ist.psu.edu/article/dube97signal.html>.

[16] Y. Huang, W. Lee, Attack analysis and detection for ad hoc routing protocols, in: Recent Advances in Intrusion Detection: 7th International Symposium, RAID 2004, 2004.

[17] Y. Zhang, W. Lee, Intrusion detection in wireless ad hoc networks, in: Proceedings of the 6th Annual International

Conference on Mobile Computing and Networking, ACM Press, 2000, pp. 275–283.
[18] Y.-C. Hu, A. Perrig, D.B. Johnson, Ariadne: a secure on-demand routing protocol for ad hoc networks, in: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking, ACM Press, 2002, pp. 12–23.
[19] Y.-C. Hu, A. Perrig, D.B. Johnson, Rushing attacks and defense in wireless ad hoc network routing protocols, in: Proceedings of the 2003 ACM Workshop on Wireless Security, ACM Press, 2003, pp. 30–40.
[20] Y.-C. Hu, A. Perrig, A Survey of Secure Wireless Ad Hoc Routing 02, 2004, pp. 28–39.
[21] The International PGPI Home Page, February 2004. <http://www.pgpi.org/>.
[22] M.G. Zapata, Internet Draft: Secure Ad hoc On-Demand Distance Vector (SAODV) Routing, 2002. <http://www.cs.ucsb.edu/~ebelding/txt/saodv.txt>.
[23] Y.-C. Hu, D.B. Johnson, A. Perrig, SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks, in: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, IEEE Computer Society, 2002, p. 3.
[24] B. Lu, U. Pooch, Cooperative security-enforcement routing in mobile ad hoc networks, in: 4th International Workshop on Mobile and Wireless Communications Network, 2002, pp. 157–161.
[25] P. Papadimitratos, Z. Haas, Secure routing for mobile ad hoc networks, in: Communication Networks and Distributed Systems Modeling and Simulation Conference, 2002, pp. 27–31. URL <citeseer.ist.psu.edu/papadimitratos02secure.html>.
[26] S. Yi, P. Naldurg, R. Kravets, Security-aware ad hoc routing for wireless networks, in: Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, ACM Press, 2001, pp. 299–302.
[27] R. Bobba, L. Eschenauer, V. Gligor, W. Arbaugh, Bootstrapping Security Associations for Routing in Mobile Ad-Hoc Networks, May 2002.
[28] Y.-C. Tseng, J.-R. Jiang, J.-H. Lee, Secure bootstrapping and routing in an IPv6-based ad hoc network, in: ICPP Workshop on Wireless Security and Privacy, 2003.
[29] J.R. Douceur, The sybil attack, in: Proceedings of the First International Peer to Peer Workshop (IPTPS 2002), 2002, pp. 251–260.
[30] P. Golle, D. Greene, J. Staddon, Detecting and correcting malicious data in vanets, in: VANET'04: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks, ACM Press, New York, NY, USA, 2004, pp. 29–37.
[31] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, K. Levitt, A specification-based intrusion detection system for AODV, in: Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM Press, 2003, pp. 125–134.
[32] T. Aura, Internet Draft: Cryptographically Generated Addresses (CGA), February 2004. <http://www.ietf.org/proceedings/04mar/I-D/draft-ietf-send-cga-05.txt>.
[33] Y.-C. Hu, D.B. Johnson, Caching strategies in on-demand routing protocols for wireless ad hoc networks, in: MobiCom'00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ACM Press, New York, NY, USA, 2000, pp. 231–242.
[34] R. Hinden, S. Deering, RFC 3513: Internet Protocol Version 6 (IPv6) Addressing Architecture, April 2003. URL <http://www.ietf.org/rfc/rfc3513.txt>.
[35] A. Tuominen, HUT AODV for IPv6 User Guide and Function Reference Guide.
[36] V. Jacobson, C. Leres, S. McCanne, TCPDUMP group's release 3.8.3, <http://www.tcpdump.org/>.
[37] T. Carstens, Programming with pcap, <http://www.tcpdump.org/pcap.htm>.
[38] M. Casado, Packet Capture With libpcap and other Low Level Network Tricks, <http://www.cet.nau.edu/~mc8/Socket/Tutorials/section1.html>.
[39] J. Morris, Perlipq: Perl extension to Linux iptables userspace queueing via libipq <http://www.intercode.com.au/jmorris/perlipq/>.
[40] E.M. Belding-Royer, Report on the AODV interop, June 2002. <http://www.cs.ucsb.edu/~ebelding/txt/interop.ps>.
[41] X. Zeng, R. Bagrodia, M. Gerla, GloMoSim: a library for parallel simulation of large-scale wireless networks, in: Workshop on Parallel and Distributed Simulation, 1998.
[42] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Mobile Computing and Networking, 1998, pp. 85–97. URL <citeseer.ist.psu.edu/broch98performance.html>.
[43] BonnMotion: A mobility scenario generation and analysis tool, March 2005. <http://web.informatik.uni-bonn.de/IV/Mitarbeiter/dewaal/BonnMotion>.
[44] Simulation Parameters for IDS simulation, May 2005. <http://ebiquity.umbc.edu/v2.1/get/a/resource/120.gz>.

Anand Patwardhan is a Ph.D. Candidate in the Computer Science and Electrical Engineering Department at UMBC. He received his Masters in Computer Science from OGI School of Science and Engineering, OHSU in 2002, and his B.E. degree in Computer Engineering from the University of Pune in 2000. His research interests are in mobile data management, security and trust in pervasive computing, and networking protocols for ad hoc networks.

Jim Parker received his B.S. degree in Computer Science from James Madison University in 1985, and his M.S. degree in Computer Science from University of Maryland, Baltimore County (UMBC) in 1998. He is currently a Ph.D. candidate in Computer Science at UMBC. As a member of the eBiquity research group at UMBC, his research focus is in the area of security as applied to mobile ad hoc computing environments.

Michaela Iorga is the President and founder of MiTech Consulting Inc Company. She has 20 years experience of computational modeling in a scientific environment with over 8 years experience in network security, personal identity verification and authentication using smart cards, wireless security, network intrusion detection, mobile device security and ad-hoc network security and role-based access control for both private and public sector organizations such as National Science Foundation, National Institute of Standards and Technology, VDG, IBM, Schaffer Corporation and Seagate. She has been involved in the development of complex security architectures, with emphasis on designing and implementing PKI-based systems, Certificate Authorities, Secure Key-Stores, Certificate Repositories, identity verification and authentication protocols using smart cards, secure wireless communication protocols, cryptographic algorithms, kernel modules, and ad-hoc routing protocols for devices with ARM, CERF and x86 architectures, running Linux OS. She holds a Ph.D. from Duke University, a Master of Science degree from "Lower Danube" University and a Bachelor degree from University of Galati, Romania. Currently, she is involved with the National Institute of Standards and Technology's Computer Security Division research projects in wireless secure communication and personal identity verification and authentication using smart cards.

Anupam Joshi is a Professor of Computer Science and Electrical Engineering at UMBC. Earlier, he was an Assistant Professor in the CECS department at the University of Missouri, Columbia. He obtained a B. Tech degree in Electrical Engineering from IIT Delhi in 1989, and a Masters and Ph.D. in Computer Science from Purdue University in 1991 and 1993 respectively. His research interests are in the broad area of networked computing and intelligent systems. His primary focus has been on data management for mobile computing systems in general, and most recently on data management and security in pervasive computing and sensor environments. He has created agent based middleware to support discovery, composition, and secure access of services/data over both infrastructure based (e.g., 802.11, cellular) and ad-hoc wireless networks (e.g., Bluetooth). He is also interested in Semantic Web and Data/Web Mining, where he has worked on personalizing the web space using a combination of agents and soft computing. His other interests include networked HPCC.

Tom Karygiannis is a principal investigator with the National Institute of Standards and Technology's Computer Security Division. As a member of the Emerging Technologies Group, he conducts research in secure electronic commerce, wireless security, network intrusion detection, mobile device security, ad hoc network security, and RFID security. He has over 18 years experience working with both private and public sector information technology organizations such as NASA's Goddard Space Flight Center, the United States Agency for International Development, UNISYS, the Johns Hopkins University Applied Physics Lab, and the European Space Agency. He holds a Ph.D. in Computer Science from the George Washington University and a Master and Bachelor of Science degree in Electrical Engineering from Bucknell University.

Yelena Yesha received the B.Sc. degree in Computer Science from York University, Toronto, Canada in 1984, and the M.Sc. and Ph.D degrees in Computer and Information Science from The Ohio State University in 1986 and 1989, respectively. Since 1989 she has been with the Department of Computer Science and Electrical Engineering at the University of Maryland Baltimore County, where she is presently an Exceptional Research Professor. In addition, from December, 1994 through August, 1999. She served as the Director of the Center of Excellence in Space Data and Information Sciences at NASA. Her research interests are in the areas of distributed databases, distributed systems, mobile computing, digital libraries, electronic commerce, and trusted information systems. She published over 140 refereed articles and also 13 books in these areas. She has received a substantial amount of research funding as PI or Co-PI from NASA, NSF, NIST, NSA, DHMH, Aether Systems, Cisco, Fujitsu, Nokia and IBM. She is a member of the editorial board of the Very Large Databases Journal, and the IEEE Transaction on Knowledge and Data Engineering, and is editor-in-chief of the International Journal of Digital Libraries. She served as general chair and program chair of several major international conferences, and recently served as the general chair of ACM SIGMOD 2005. During 1994, she was the Director of the Center for Applied Information Technology at the National Institute of Standards and Technology. She is a senior member of IEEE, Fellow of IBM CAS and a member of the ACM.