Application-Layer Intrusion Detection in MANETs

Katharine Chang and Kang G. Shin
The University of Michigan, Ann Arbor, MI 48109-2121
{katchang, kgshin}@eecs.umich.edu

***Abstract*— Security has become important to mobile ad hoc networks (MANETs) due mainly to their use for many mission- and life-critical applications. However, the broadcast nature of inter-node communications and node mobility in MANETs make it very challenging to secure MANETs. Moreover, their constantly-changing topology causes network node density and neighbor relationships to change dynamically. This paper presents an intrusion detection system (IDS) for MANETs at the application layer. The IDS utilizes (1) both anomaly and misuse detection schemes to identify attacks in MANETs and (2) mobile agents (MAs) to augment each node's intrusion-detection capability. In particular, each node is equipped with a local IDS, and MAs will be dispatched periodically or on-demand to augment each node's IDS. We present the design of this IDS and the overall network structure, as well as the methods for authenticating and dispatching MAs. We also evaluate the trade-offs between different design parameters of MAs.**

I. Introduction

A mobile ad hoc network (MANET) is built with mobile nodes which communicate via wireless radio links. Instead of using a central base station for nodes to communicate with one another, MANETs do not rely on any pre-defined infrastructure and operate in peer-to-peer mode. Nodes within the communication range communicate via wireless radio links, and for those outside the communication range, use other nodes to relay their packets. Mobile nodes may move away from their current locations and re-join the network from different locations in the network, thus dynamically changing their network topology and node density. MANETs are being developed and deployed for many mission- and life-critical applications such as military tactical operations (e.g., future combat system (FCS)), emergency search-and-rescue missions, and mobile tele-conferencing. However, the dynamically-changing topology of MANETs make them vulnerable to various attacks.

In recent years, security has become a primary concern to the communications in MANETs. Unlike wired networks, security in wireless networks is difficult to achieve due to the broadcast nature of inter-node communications. In MANETs, it is even more difficult to achieve security because of node mobility and constantly-changing group membership. Intrusion prevention is not guaranteed to work all time either, and can only combat outsider attacks. Therefore, we need a strong intrusion detection system (IDS) which plays a critical role in securing MANETs. An IDS can discover malicious activities or insider attacks mounted by compromised nodes in the network. The IDS then tries to prevent intrusions that compromise system security, and upon detection of an intrusion, it tries to recover from the damages inflicted by the intrusion. Considering continuous discovery of new vulnerabilities, the IDS must be effective and efficient in identifying attacks, and then neutralizing them.

The traditional IDSs developed for wired networks are difficult to use for MANETs because of their architectural differences. Without centralized audit points like routers, switches, and gateways, MANETs can only collect audit data locally and thus require a distributed and cooperative IDS. Other differences between wired networks and MANETs include traffic patterns, node mobility, and node constraints. These differences all render the traditional IDSs hard to be directly applied to MANETs. Nodes in MANETs can move freely through the network, and thus their dynamically-changing network topology makes MANETs very different from the traditional wired networks. Also, nodes in MANETs usually have slower communication links, limited bandwidth, limited battery power, and limited memory. Therefore, these constraints make the design of IDS in MANETs much more challenging than in wired networks.

Due to the dynamically-changing topology of MANETs, neighbor relationships and node density vary with time. For MANETs with high node mobility, it is very difficult to design an IDS that is distributed and light-weighted, and consists of cooperative nodes in physical proximity. To meet this challenge, we propose an MA-based application-layer IDS for MANETs. It utilizes both anomaly and misuse detection to identify attacks and also utilizes MAs to augment each node's intrusion detection capability. Our goal is to detect and prevent viruses, worms, and malicious applications on each node by using the MA technology to complement the IDS.

The main contribution of this paper is the use of MAs to augment the application-layer IDS in MANETs which is a significant departure from most existing IDSs in MANETs that target the network layer. Our application-layer IDS uses system-call sequences to detect intrusions on each node. It also uses MAs to augment the IDS by updating attack signatures and normal application profiles, and patching and installing (new) programs on each node. MAs can also augment the detection capability by being dispatched for further analysis and diagnosis on network nodes when an anomaly is detected. Finally, MAs can be dispatched to verify the correctness of IDS agents. Another contribution of this paper is the mechanism for dispatching MAs to network nodes for update, analysis, and verification.

The remainder of this paper is organized as follows. We first list the advantages of using MAs in IDSs in Section II. The system architecture is then presented in Section III along

with the assumptions and the attack model used in this paper. Section IV details the design of our MA-based IDS, and the use of MAs in MANETs. Section V analyzes the security of the proposed IDS while Section VI evaluates its performance. We then discuss the related work in Section VII. Finally, the paper concludes with Section VIII.

II. Why are MAs Needed for Intrusion Detection?

Since the mobile nodes in MANETs are energy-constrained, we need to design protocols that are lightweight and energy-efficient. MAs offer many advantages [1] when used in an IDS, and will help overcome the difficulty of building distributed systems and protocols. Below we list and describe the advantages of using MAs for intrusion detection.

Reducing Network Load: MAs transfer the computation and detection function to the network nodes with audit data instead of transmitting large amounts of audit data to the servers for computation and detection, thus reducing the network load.

Overcoming Network Latency: MAs can be dispatched from the servers to network nodes to detect malware and take corrective actions in real time. The MAs can operate directly on the nodes and respond faster to a potential intrusion than communicating with the servers for assistance.

Making the IDS Attack-Resistant: MAs can be used in the IDS to avoid single-point-of-failures. The time of an MA's arrival at each node, the reporting mechanism, and the detection algorithm the MA uses are made unpredictable so that attackers may not know this information.

Autonomous Execution: MAs can continue to function even when portions of the IDS or the network get destroyed or malfunction. MAs can increase the IDS's fault-tolerance by operating independently of the platform.

Dynamic Adaptation: MAs have the ability to sense the execution environment and react to changes. Also, MAs can adapt to the environment as they can be retracted, dispatched, or put to sleep as the network and host conditions change.

Platform Independence: MAs can operate in heterogeneous environments by having a virtual machine or interpreter on the host platform. This capability makes a perfect fit for MANETs as nodes in the network typically are comprised of many different computing platforms.

Upgradability: MAs can perform program updates, and anomaly and misuse detection on each node. MAs can carry the most up-to-date program patches, normal application profiles, and attack signatures to the nodes for upgrade while the IDS keeps working on each node.

Scalability: MAs help distribute the computational load to different nodes in the network instead of having all the computation processed on the servers, and reduce the network load. This advantage enhances scalability and makes the IDS more fault-resistant.

III. Assumptions, Attack Model, and System Architecture

We first state the assumptions we use, then present the attack model, and finally, describe the system architecture of our IDS design.

A. Assumptions

- A1. A MANET is composed of a large number of mobile nodes and one secure stationary MA server.
- A2. Mobile nodes and the MA server in the network have unique node IDs.
- A3. A trusted offline certificate authority (CA) is used to bind public keys with respective mobile nodes and the MA server. Each node has a pair of private and public keys, and the public key can be known to other nodes via the certificate issued by the CA.
- A4. The MA server and the mobile nodes have a shared symmetric key for message encryption. All messages between the MA server and the nodes in the network are encrypted with this symmetric key.
- A5. Before deployment, mobile nodes in the network have agreement with the MA server about the security policies for the authorized actions an MA can perform on a node.

B. Attack Model

Security attacks in MANETs can be categorized into passive or active attacks. Passive attacks include eavesdropping of data, and traffic analysis and monitoring. Active attacks include replication, modification, insertion and deletion of data to be exchanged, external service attacks, resource consumption (e.g., DoS attacks), and physical attacks. Security attacks can also be categorized according to protocol layers. Some attacks on the application layer are data corruption, repudiation, application abuses, DoS attacks, and mobile virus and worm attacks. There are also some attacks that target across multi-layers, such as DoS attacks, impersonation, and man-in-the-middle attack.

In this paper, we focus on detecting and defending against application-layer attacks mentioned above. Our goal is to detect malicious applications and mobile virus and worm attacks on the nodes in the network.

C. System Architecture

We consider a network of a large number of mobile nodes and one secure stationary MA server. There can be multiple MA servers deployed in the network to avoid single-point-of-failures. However, it will require coordination and interactions between the MA servers, so we focus on using one MA server in the following design. The MA server is used for managing MAs, normal application profiles, and attack signatures generated by MAs in the network, and is deployed in the network along with the mobile nodes. The MA server is akin to the command and control (C&C) center in a battle field. The C&C center can issue orders to the troops in the battle field. The MA server acts like the C&C center and dispatches MAs to the nodes in the network when needed. It will periodically broadcast beacon messages for nodes in the network to locate itself. When a node needs an MA for assistance, it will request an MA from the MA server. The proposed IDS architecture for MANETs is depicted in Figure 1.

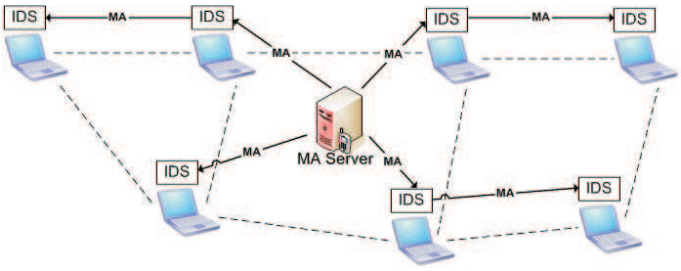

Fig. 1. The IDS system architecture for MANETs.

Each mobile node has its own local IDS which is responsible for monitoring and detecting attacks, and for responding to the attacks detected. The local IDS performs anomaly and misuse detection. The misuse detection is used to detect known attacks on the node, while the anomaly detection is used for the detection of new or previously-unknown attacks. MAs are designed to update attack signatures and normal application profiles, patch and install programs, analyze and diagnose anomalous nodes, and verify the local IDS agents.

Each IDS consists of three agents: the monitoring and detection agent, the response agent, and the secure communication. There is also a local database in each node for storing system audit data, attack signatures, normal application profiles, and the IDS logs. For the execution of MAs, there is a mobile agent place on each node. The local IDS architecture in each mobile node is shown in Figure 2.

The monitoring and detection agent monitors the application-level activities and system calls on each node, and also compares the monitoring activities with the attack signatures and normal application profiles stored in the node's local database. Once a malicious activity is detected by the misuse detection via signature matching, a proper response will be formulated by the response agent to recover the node from the damages occurred to it. If the monitoring and detection agent detects intrusion by anomaly detection via an above-threshold deviation from the normal profile but no signatures match the attack, then the response agent will request the MA server to dispatch an MA for further analysis and diagnosis. The secure communication component is used for the mobile nodes to securely communicate with the MA server and other nodes in the network.

IV. Design of Application-Layer IDS

As mentioned earlier, our goal is to design an application-layer IDS that utilizes anomaly and misuse detection to identify malicious applications as well as mobile virus and worm attacks. MAs are utilized to augment each node's capability of intrusion detection in the network. We assume the existence of the MA server for managing and dispatching MAs to nodes in the network. Recall that each node in the network has its own local IDS.

Described below is the detailed design of using MAs for the application-layer intrusion detection.

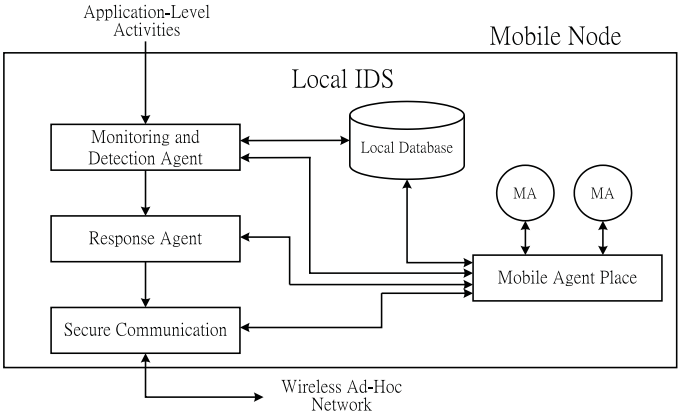

Fig. 2. The local IDS architecture on a mobile node.

A. Local IDS for Intrusion Detection

We use misuse detection as host-based intrusion detection of known attacks in each node. The monitoring and detection agent in the IDS audits the application-level activities and system calls in each node and compares them against the attack signatures stored in the node's database that represent worm and virus signatures, as well as misbehavior signatures of applications. The behavior-based attack signatures capture an incorrect order of instructions or incorrect control flow of programs, unlike traditionally-used signatures generated from raw byte sequences in malicious executables and takes up less storage space in the database.

The MAs dispatched to the mobile nodes can generate new attack behavior signatures, usually after the identification of an intrusion via anomaly detection, and the confirmation of the existence of intrusion after performing analysis and diagnosis of the nodes. The MAs will report these newly-generated attack signatures to the MA server, and the MA server will dispatch MAs to other nodes in the network to add the new attack signatures. For misuse detection, the MA server is used to store and update the attack signatures collected from the nodes in the network.

To use anomaly detection in our local IDS, a normal profile is computed for each application program using its audit data on the MA server. As we are targeting the detection of application-layer intrusions, the audit data collected on the MA server is the application-level activities and system calls invoked by the application programs. Specifically, the sequence of the system calls generated by each application program and collected in the audit data is used to compute the normal profiles.

The nodes will first be deployed in the network with the normal application profiles generated by the MA server for existing applications. When a new normal application profile has been generated due to a program patch or new application installation, MAs will carry the new profile to nodes in the network for update.

The monitoring and detection agent will compare the audit data of system traces on each node with the normal application

profiles. The difference between a sequence of system calls against a normal application profile can be computed using Hamming distance. Any major deviation of abnormal activities from the normal application profiles will be detected and can be used as an alert to the local IDS. Once an anomaly is detected, the response agent will request an MA from the MA server for further analysis and diagnosis on the node. If the MA confirms the attack, then a proper response action will be taken at the node and a new attack signature will be generated for the detected attack.

Each node also logs all the IDS-related activities including both misuse and anomaly detection histories for MAs to analyze and report to the MA server. In addition to intrusion detection, MAs can also collect and analyze the audit data and IDS logs stored in local nodes. The collected information will be reported to the MA server and used for further analysis.

B. MA Functions

The MA server creates and dispatches three types of MAs: (1) update MAs, (2) analysis MAs, and (3) verification MAs. The update MAs are dispatched as needed, the analysis MAs are dispatched upon request by nodes, and the verification MAs are dispatched periodically. The three types of MAs are detailed next.

1) The Update MA:

The update MAs are used by the MA server to add new attack signatures and normal application profiles, and patch and install programs on the mobile nodes. When the MA server receives new attack signatures generated by the analysis MAs, it will dispatch update MAs carrying the signatures to the nodes in the network. All nodes need to be updated with these new signatures for effective and up-to-date misuse detection.

If a vulnerability is detected and application programs on the nodes need to be patched, or new application programs are being installed, then the MA server will also dispatch to the nodes the update MAs carrying the patches and new programs. When a program is patched, then the normal application profile needs to be updated, or if a new program is installed, then the nodes also need to have a normal profile for the new application. The MA server will dispatch the update MAs along with the new normal application profiles generated by the MA server to the nodes for update.

The update MAs are dispatched to the nodes when needed, so the nodes need not request updates from the MA server. This will greatly reduce the number of requests from nodes and prevent the MA server from becoming a communication bottleneck. With the update MAs dispatched to the nodes, the MA server need not send the same updates to different nodes, and thus, the network load will be reduced significantly.

2) The Analysis MA:

The analysis MAs are dispatched to the requesting nodes for further analysis and diagnosis for anomaly behaviors on them. When the monitoring and detection agent in a node's local IDS detects an anomaly but the anomaly did not match any attack signature in its database, then the response agent in the IDS will send an anomaly report to the MA server and request an analysis MA from the MA server for further investigation. The anomaly report includes the related IDS logs and the intrusion information about the anomaly detected on the node. Depending on the content of the anomaly report, the MA server will choose the most suitable analysis MA. The analysis MA is capable of a more detailed analysis and diagnosis than the local IDS, and can evaluate the detected anomaly behavior and determine if it is an intrusion.

If the analysis MA determines the anomaly behavior not to be an intrusion, then it will send a detection report to the MA server and destroy itself. However, if the anomaly behavior is determined as an intrusion, then the analysis MA will start the intrusion response through the response agent on the local IDS. Finally, the analysis MA will create an attack signature of the newly-identified intrusion and report the new attack signature to the MA server. After completing the response for the intrusion, the analysis MA will send a detection report to the MA server and destroy itself.

Note that if the analysis MA cannot determine if the anomaly is an intrusion or not, then it can request a different analysis MA from the MA server for help. The analysis MA can also migrate to neighbor nodes to perform further investigation. The investigation at the neighbor nodes is the multi-point network-based anomaly detection. The analysis MAs can determine new attacks by analyzing and diagnosing both on a node and on its neighbor nodes. If the analysis MA still cannot (un)confirm an anomalous behavior, it will then send the relevant IDS logs and audit data back to the MA server for further analysis.

The analysis MAs are dispatched to the nodes requesting for further analysis of anomaly behaviors. The reason for sending analysis MAs to the nodes instead of having the nodes send all the audit data, IDS logs, and related information to the MA server for analysis is to reduce the network load. Also, the MAs can overcome network latency as the analysis MAs can be dispatched from the MA server to perform analysis on nodes in real time. The analysis MAs can be executed on the nodes and can respond faster instead of needing to communicate back and forth with the MA server for assistance.

3) The Verification MA:

The verification MAs are periodically sent by the MA server to verify the IDS agents on the nodes and check on the IDS logs and the local IDS execution states. These MAs are used to prevent the local IDS from being compromised by attackers. Similar to the program integrity verification in PIV [2] and SWATT [3], we use hash verification to determine the integrity of the IDS agents on the nodes.

The MA server will periodically send the verification MA with a randomly-generated hash key to the network. The nodes that received the verification MA must execute it to check the integrity of the IDS agents in its local IDS. Since the MA server keeps copies of the IDS agents, we can use the verification MA to verify the integrity of these agents. When the verification MA is executed, it computes a hash over the IDS agents using the hash key it carries, and the hash value will be sent back to the MA server for verification.

The verification MA will also check the IDS logs and see if there is any anomaly or unreported events. If a node fails the verification, then the MA server will either send an update MA to correct the IDS agents or shut down the entire node.

C. MA Authentication and Authorization

To authenticate MAs and nodes in the network, we use the public key infrastructure (PKI). We assume there is a trusted offline certificate authority (CA) that issues certificates to the MA server and nodes in the network. The certificates contain the public key and the ID of the owner node and let other nodes in the network verify the owner node's credential. The CA also issues each node a corresponding pair of private and public keys. When the MA server dispatches MAs to the network, it will include its certificate issued by the CA and sign the MAs with its private key for authentication.

When an MA is dispatched to the network and needs to send detection information back to the MA server, it will execute an encrypted function and secretly sign the detection reports for the MA server [4]. The MA carries a program to nodes in the network that implements an encrypted function for the digital signature. Before sending a detection report back to the MA server, the MA will execute the encrypted function on the local node to sign the report and attach the signature along with the report. When the detection report reaches the MA server, the MA server can check the signature to verify the authenticity of the report.

The complete authentication steps are described as follows.

S1. The CA issues the MA server S and each mobile node a pair of private and public keys, and a certificate that contains the public key and ID of the node. S and mobile nodes share a symmetric key K.

S2. Each update and verification MA MA_S sent by S carries S's certificate $Cert_S$, and S encrypts MA_S with K and signs $\{MA_S\}_K$ with its private key k_S.

S3. Mobile node A that receives $\{MA_S\}_K$ from S verifies $Cert_S$ and gets S's public key K_S from $Cert_S$. If the $Cert_S$ verification is valid, A will then verify S's signature with K_S. If the verification of the signature is valid, then $\{MA_S\}_K$ will be decrypted using K and then executed. MA_S will be deleted if any of the verification fails.

S4. If MA_S is an update or verification MA, then MA_S will be sent to another mobile node B for update or verification. The previous steps are repeated until MA_S expires and is deleted.

S5. If there is an anomaly detected on A by the local IDS, A will send an anomaly report $AReport_A$ to S requesting for an analysis MA. A will encrypt $AReport_A$ with S's public key K_S, and then sign $\{AReport_A\}_{K_S}$ with A's private key k_A for authentication. Finally, A sends the encrypted report $\{AReport_A\}_{K_S}$, the signature, and its certificate $Cert_A$ to S.

S6. When S receives $\{AReport_A\}_{K_S}$ from A, it will fist verify $Cert_A$ and get A's public key K_A. If the $Cert_A$ verification is valid, S will then verify A's signature with K_A. Finally, S can decrypt $\{AReport_A\}_{K_S}$ with its private key k_S if the verification succeeds. However, $AReport_A$ will be discarded if any of the verification fails.

S7. S will send analysis MA MA_S to A as described in S2, and A can authenticate MA_S as described in S3. After MA_S is executed on A, MA_S will need to send a detection report $DReport_{MA_S}$ to S for a status report or request for help. A will first encrypt $DReport_{MA_S}$ with S's public key K_S, then execute MA_S to get the digital signature DS_1 for $\{DReport_{MA_S}\}_{K_S}$ using the encrypted function carried by MA_S. A then signs $\{DReport_{MA_S}\}_{K_S}$ with its private key k_A to get digital signature DS_2 and combine it with DS_1. Finally, A sends the encrypted report $\{DReport_{MA_S}\}_{K_S}$, the signatures $DS_1 + DS_2$, and its certificate $Cert_A$ back to S.

S8. When S receives $\{DReport_{MA_S}\}_{K_S}$ from A, it will first verify $Cert_A$ and get A's public key K_A. If the $Cert_A$ verification is valid, S will then verify $DS_1 + DS_2$ with K_A and the encrypted function. Finally, S can decrypt $\{DReport_{MA_S}\}_{K_S}$ with its private key k_S if the verification succeeds. However, the encrypted report $\{DReport_{MA_S}\}_{K_S}$ will be discarded if any of the verification fails. Depending on $DReport_{MA_S}$, S will then decide whether to send more verification MAs to A for intrusion detection and response or to process and update the reported information received.

As for MA authorization, we assume that before deployment, mobile nodes in the network have an agreement with the MA server about the security policies for the authorized actions an MA can perform on the nodes. The MA server should only authorize legal actions for MAs to perform on each node. Before an MA can be executed on each node, it will need to pass the authentication. After the authentication, however, if the MA attempts to perform actions that are not on the authorized list, the node will disallow the MA to take them.

D. MA Dispatching

As described in Section IV-B, there are three types of MAs: the update MAs, the analysis MAs, and the verification MAs. The update MAs are sent as needed, the analysis MAs are sent upon request, and the verification MAs are sent periodically. The MA server dispatchs MAs and controls the generation of MAs, their quantities, and their communication timing.

The time interval for periodic verification MAs to be sent to the network can vary. Depending on the condition and state of each network area, the MA server can decide adaptively how often to dispatch verification MAs to a certain area to verify the local IDSs. However, the MA server is susceptible to DoS attacks from malicious nodes that request analysis MAs or send false detection reports. We can maintain packets' path histories and determine the source of attacks. We can

also restrict the number of times a node can contact the MA server within a certain time duration. For confidentiality and authentication purposes, we use the network-wide symmetric key for encryption and PKI for digital signature, as stated in Section IV-C.

When MAs are dispatched to the network, they send back two types of reports to the MA server: periodic and detection reports. The periodic reports are for fault-tolerance and data collection. In case the MA server didn't receive the periodic reports from an MA, it will check for the MA or send more MAs to continue the MA's task. The update and verification MAs need to periodically send periodic reports back to the MA server. The periodic reports include the nodes the MA visited, the nodes' battery information, and the nodes' update/verification results including the related IDS information. The detection reports are sent back to the MA server by the analysis MAs to report the analysis result for the anomaly behaviors detected, to request more analysis MAs from the MA server, or to report relevant IDS logs, audit data, and intrusion evidence to the MA server for further analysis. The detection reports include where the anomaly behavior is detected, the related IDS logs for the anomaly, and if the anomaly is confirmed and countermeasures applied.

Each MA has a time-to-live (TTL) parameter set by the MA server that determines the number of nodes the MA can visit in the network. After each MA visits a node, the TTL will be decremented by one. When the TTL on the MA expires, the MA destroys itself.

The MA-dispatching protocol is summarized as follows.

1) When an MA server S dispatches MA_S to the network, it includes its certificate $Cert_S$ and encrypts MA_S with the network shared symmetric key K and signs MA_S with its private key k_S for authentication as described in Section IV-C.
2) If MA_S is an update MA, then S includes in MA_S the new attack signatures it collected from the network for signature update. S also includes any program patch or new program in MA_S for program update, and the newly-generated normal application profiles.
3) If MA_S is an update MA or a verification MA, then S will dispatch MA_S to a network area for random traverse. If MA_S is an analysis MA, then it will be dispatched to the requesting node.
4) Each MA will have a TTL parameter that determines the number of nodes the MA can visit in the network.
5) While MA_S travels in the network, it sends back periodic or detection reports to S to report its update/verification or diagnosis results on each node. The encryption and authentication details are described in Section IV-C.
6) An MA destroys itself upon expiration of its TTL.

After its execution on one node, the update and verification MA will move to other nodes in the network for update/verification, and the analysis MA can decide whether to migrate to neighbor nodes for further analysis and diagnosis or to destroy itself after it finishes the analysis. The MA server can send multiple MAs to different network areas. The more MAs sent out, the faster the update/verification will be done. If fewer MAs are dispatched, then it will take longer for MAs to visit other nodes and finish their tasks. Dispatching too many MAs will incur more traffic as the same MA may visit a node multiple times. We would like to limit the amount of communications between the MA server and the nodes in the network. Therefore, we need to find an optimum number of MAs to be dispatched to the network.

1) Message-Loss/Compromise Problems: Since MAs need to travel multiple hops through the network to reach certain nodes or send periodic and detection reports back to the MA server, we need to consider message-loss/compromise problems that might occur at the intermediate relay nodes. We use secure routing [5] to send MAs to the network and periodic and detection reports back to the MA server.

If MAs are lost or captured by malicious intermediate nodes, nodes in the network will not receive the periodically sent verification MAs or the requested analysis MAs. If nodes have not received MAs for longer than a certain time $Time_{MA}$, then they can request MAs from the MA server again.

In case periodic reports are lost or captured, the MA server will not receive the periodic reports from MAs nor know the MA status and the update/verification results from the reported information. If the MA server has not received periodic reports from an MA for longer than a certain time $Time_{report}$, then the MA server will look up the previous periodic reports the MA returned and then mark the visited/routing nodes as suspicious nodes. The MA server will then send query messages to the suspicious nodes to look for the MA and request periodic reports from the MA. If still no periodic reports are received by the MA server, then it will send analysis MAs to the suspicious nodes for anomaly detection. If the detection reports are lost or captured, then the MA server can send more analysis MAs to the node under investigation.

E. Intrusion Response

When an intrusion is detected on a node either by misuse detection (via attack signature matching) or by anomaly detection (via major deviation from a normal application profile), the response agent in the node's local IDS will respond to the detection. The response depends on the degree of damages done by the intrusion, the type of intrusion, and the type of the malicious application. When an anomaly is detected, the response agent will ask the MA server for an analysis MA for further analysis and proper intrusion response.

The response agent can try to re-program or disinfect the compromised node if the damage caused by the intrusion can be fixed by re-programming the node. Or, the response agent can ask the MA server for program patches (using update MAs). The response agent can also send notifications and alerts to the network for re-authentication, or exclude and shut down the compromised nodes.

V. Security Analysis

We first discuss the security of the MA-based IDS when MAs and nodes are compromised, and then analyze the

security of the MA-based IDS protocol against various attacks.

A. Defense Against Compromised MAs and Nodes

An attacker must first be able to fake the signature of an MA with the MA server's private key before he wants to compromise the MA. Only when the attacker can get the compromised MA to be signed, will the MA pass the authentication and be executed on nodes in the network. Even if the attacker has successfully faked the signature and the compromised MA has passed the authentication but tries to perform illegal actions that are not in the authorized list of the node, then the node will protect itself by disallowing the MA to execute the unauthorized actions.

It might be possible for the compromised update MAs to carry the faked normal application profiles and attack signatures to the nodes, and let the node's IDS detect intrusions even when the node actually behaves normally. The IDS response agent will then respond to the attack and request analysis MAs from the MA server for further analysis and diagnosis. The analysis MAs will be dispatched to the node and will aid its detection and response to the intrusions. The compromised update MA can then be detected when the analysis MAs arrive and analyze the IDS logs on the node.

As for a compromised node that has subverted its local IDS, the verification MAs will arrive at the node periodically and verify the integrity of the IDS agents on the node. The compromised IDS will be detected if the IDS agents have been modified, or the IDS logs and execution states are incorrect. Note that a compromised node has to execute the verification MA; otherwise, the verification MA will suspect the node to have been compromised.

If the compromised node can successfully pass the verification and alter its IDS logs, then it must also pass the analysis performed by the analysis MAs that may be sent to the node for further investigation. Therefore, for a compromised node to go undetected by the analysis MAs, it cannot perform abnormal activities during the MA's diagnosis. If the node discards the analysis MA and hence no detection reports are delivered to the MA server, then the MA server will dispatch more analysis MAs for further detection. Also, when the neighbor nodes detect the abnormal behaviors from the compromised node, they will request analysis MAs, and the compromised node can be excluded from the network when all the nodes are required to be re-authenticated.

B. Defense Against Various Attacks in Ad Hoc Networks

We now describe how the MA-based IDS protocol can defend against various attacks in MANETs.

1) Defense Against Passive Attacks:

When an MA is to be dispatched by the MA server, it is encrypted with the network-wide shared symmetric key, along with the signature signed by the MA server's private key. The encryption is to prevent attackers from eavesdropping the network and seeing the content of the MAs. However, if one node is compromised and the shared key is revealed, then it is necessary to renew the shared key among the nodes in the network. We can also group nodes depending on the geographical areas of the network and assign different group keys to different groups. When a node leaves or joins a group, the group key will need to be renewed or distributed securely to the joining node [6], [7].

As for the periodic and detection reports that the MAs send back to the MA server, they are all encrypted with the MA server's public key, so only the MA server can decrypt the reports. The reports are also signed by the MA's signature function and the node where the MA resides. The MA digital signature function is encrypted, so neither the attacker nor the node can see the signature function. Therefore, the attacker will not get the contents of the periodic and detection reports by simply eavesdropping on the network.

2) Defense Against Active Attacks:

To prevent an attacker from spoofing or inserting false data, we sign every MA, the periodic and detection reports, and the anomaly reports from nodes with the MA server's or the nodes' private keys to achieve authenticity and integrity. We also encrypt the MAs with the network-wide symmetric key. The MAs also carry an encrypted function for digital signature to ensure the authenticity of the periodic and detection reports.

The node compromised by an attacker can be detected by the local IDS, and the response agent in the IDS will handle the intrusion. If the local IDS is compromised, then the periodically-sent verification MAs will be able to detect the faulty IDS agents, and the MA server will dispatch analysis MAs for diagnosis and response.

Malicious nodes can cause service disruption and Denial-of-Service (DoS) attacks. There is not an easy way to prevent such nodes from launching attacks, but they can be detected and then removed from the network. We rely on the local IDS at each node to detect the nodes' malicious behaviors. Since we let a local IDS monitor and detect known intrusions and anomalies on each node, and let MAs aid the detection, once a node identifies a malicious or anomaly behavior, its IDS response agent can evict the compromised node from the network and the neighbor nodes will ignore any messages from the compromised node.

Another potential attack is for a node to launch DoS attacks to the MA server, requesting MAs from or sending reports to the MA server. This kind of DoS attacks targeting the MA server can be handled by having the MA server keep path histories of the messages sent to it in order to pinpoint the attacker and restrict the number of times a node can contact it within a certain time duration.

Sybil attacks [8] are particularly harmful in MANETs where a Sybil node illegitimately fakes to have multiple identities in the network. Our MA-based IDS withstands such attacks since each node will need to have a private key and a matching certificate to authenticate its identity. Since each node will have a preloaded private key and certificate, no node can generate the private key and certificate, and pretend to be another node without compromising the node.

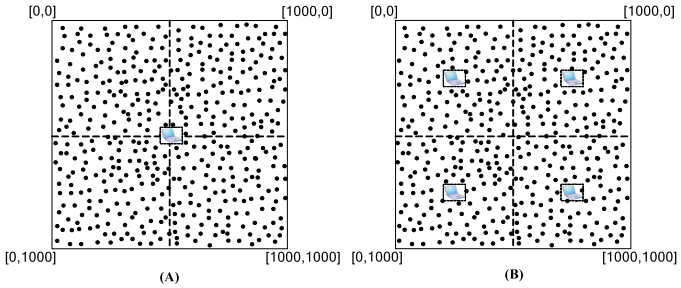

Fig. 3. The simulation environment. Laptops represent the MA servers and black circles represent mobile nodes in the network.

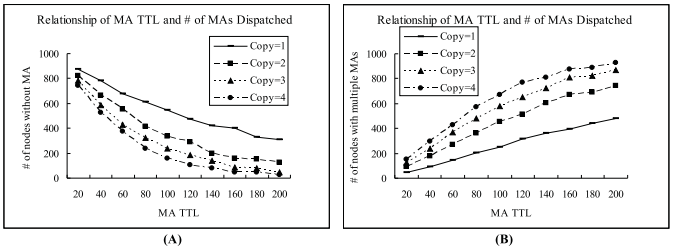

Fig. 4. The relationship between different MA TTLs and numbers of MAs dispatched by one MA server using the simulation environment in Figure 3(A) (including the extra sent MAs). The number of nodes that did not receive any MA is shown in (A) and the number of nodes that received multiple MAs is shown in (B).

VI. Evaluation

We use simulation to evaluate the trade-offs of different design parameters in the MA-based IDS. We developed our own simulator using C. We first simulate with different MA TTLs and numbers of MAs dispatched by the MA server to the network to learn the number of nodes in the network that did not receive any MA. Our goal is to have as few nodes in the network as possible that did not receive the dispatched MAs. In this simulation, we study the trade-offs between dispatching more MAs to the network or allowing the MAs to travel more hops in the network and have larger TTLs. We can also study the number of nodes in the network that have received multiple MAs during the dispatching. By studying the number of nodes that did not receive MAs and the number of nodes that receive multiple MAs, we can decide the suitable number of MAs to be dispatched and the appropriate MA TTL value.

We also simulate the deployment of multiple MA servers in the network, and see how it affects the MA distribution. In this simulation, we study the trade-offs between positioning the MA servers at random or uniformly positions. And we simulate the deployment of more MA servers in the network.

A. Simulation Setup

We first simulated the MA-based IDS with randomly-generated networks consisting of 1000 nodes and an MA server in a 1000×1000 units^2 area. Nodes and the MA server are assumed to have a communication range of 75 units. The MA server is deployed at the center of the network at (500, 500). On average, the MA server has 15–20 neighbor nodes. This simulation environment is depicted in Figure 3(A).

We simulated and compared the number of MAs dispatched by the MA server with different MA TTLs. By changing the MA TTLs and the number of MAs dispatched, we examine the number of nodes that did not receive the MAs dispatched from the MA server and the number of nodes that received multiple MAs. In the simulation, the MA server will first dispatch some MAs to its neighbor nodes in the network. After receiving and executing the MA, the MA server neighbor node will forward the MA to another node, and this is repeated until the MA's TTL expires, at which time the MA destroys itself.

Next we simulated the case with the deployment of four MA servers in the network. The four MA servers are deployed at (250, 250), (750, 250), (250, 750), and (750, 750), respectively. On average, each MA server has 15–20 neighbor nodes. The simulation environment is depicted in Figure 3(B). While changing the MA TTLs and numbers of MAs dispatched, we studied and compared the MA distribution results with only one MA server in the network. We also simulated the case with the random deployment of 4 and 5 MA servers in the network and compare the MA distribution results. With MA servers are randomly deployed, each MA server has on average of 10–20 neighbor nodes.

B. Results

The MA-based IDS is simulated with 10 different MA TTL values, ranging from 20 to 200, with 20-hop increments. The network can be separated into four quadrants, with one MA server located at the center of the network, as shown in Figure 3(A). Since, on average, the MA server has 15–20 neighbor nodes, it will dispatch 10 MAs by sending them to ten of its randomly-selected neighbors. After receiving the MAs from the MA servers, the nodes will execute the update or verification MA, then forward the MA to their randomly-selected neighbor nodes. Note that a node will not forward the MA to the node where it received the MA from, unless the node has only one neighbor.

If the MA server dispatches only 10 MAs with TTL=20, then at most 200 nodes would have received the MAs. There are, however, 1000 nodes in the network and those that have not received MAs then need to request MAs from the MA server. We further let the MA server sends extra MAs to its neighbors and let them forward the MAs without executing them. We simulated the system while the MA server sends 0 extra MA (Copy=1), 1 extra MA (Copy=2), 2 extra MAs (Copy=3), and 3 extra MAs (Copy=4), respectively.

The simulation results for different MA TTLs and numbers of MAs dispatched by the MA server (including the extra sent MAs) are plotted in Figure 4. The four curves represent different numbers of MAs dispatched by the MA server. The Copy=1 curve represents the case of the MA server dispatching 10 MAs, whereas the Copy=4 curve represents the case when each MA server dispatches 40 MAs. As shown in the four curves in Figure 4(A), the more MAs dispatched, the less nodes in the network are left without MAs visiting them. The

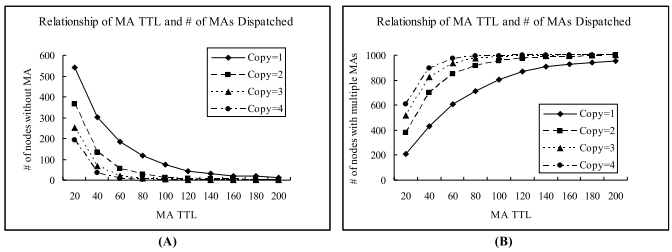

Fig. 5. The relationship between different MA TTLs and numbers of MAs dispatched by four uniformly positioned MA servers using the simulation environment in Figure 3(B) (including the extra sent MAs). The number of nodes that did not receive any MA is shown in (A) and the number of nodes that received multiple MAs is shown in (B).

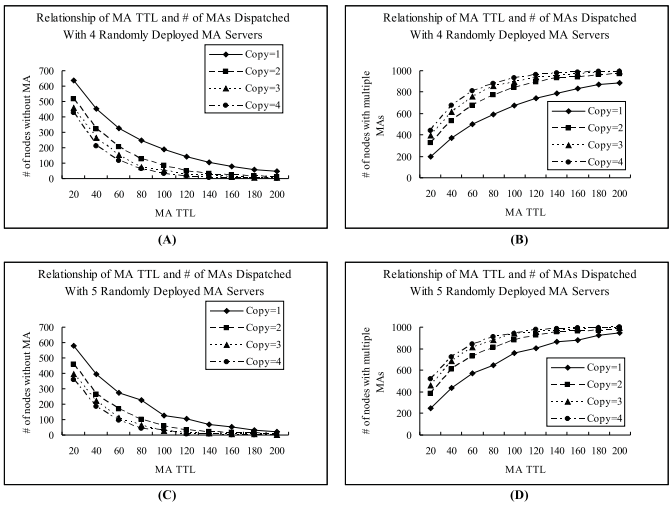

Fig. 6. The relationship between different MA TTLs and numbers of MAs dispatched by randomly-deployed MA servers (including the extra sent MAs). With four MA servers deployed, the number of nodes that did not receive any MA is shown in (A) and the number of nodes that received multiple MAs is shown in (B). With five MA servers deployed the number of nodes that did not receive any MA is shown in (C) and the number of nodes that received multiple MAs is shown in (D).

MA TTL values also affect the number of nodes receiving MAs in the network. The larger the MA TTL value, the more nodes the MA can visit during its lifetime. Therefore, the larger the MA TTL value, the less nodes in the network are left without MAs visiting. However, as shown in Figure 4(B), the more MAs dispatched, the more nodes in the network will have more than one MA being dispatched to them. Also, the larger the MA TTL value, the more nodes will have more than one MA being dispatched to them.

From Figure 4, we can see a trade-off between minimizing the number of nodes in the network without MAs and the number of nodes that have more than one MA dispatched to them. The goal is to minimize the number of nodes in the network without MAs, but at the same time, we should not have too many nodes with the same MAs dispatched to them. There is also a trade-off between whether to dispatch more MAs to the network or to make the MA TTL larger. The more MAs are dispatched, the more traffic and communication overheads will be incurred. However, the larger the MA TTL, the longer the MAs will live, the longer it takes for all nodes to receive the MAs and the higher risk for the MAs to be captured or compromised.

The simulation results for four MA servers deployed in the network, with one MA server located at the center of each quadrant as shown in Figure 3(B) are plotted in Figure 5. From Figure 5, we can see that when the MA TTL becomes 40, both Copy=3 and Copy=4 curves have less then 100 nodes that have not received MAs (65 nodes when Copy=3 and 35 nodes when Copy=4). This is acceptable for a network of 1000 nodes. Again, there is a trade-off between whether to dispatch more MAs in each period or to make the MA TTL larger.

From a comparison of Figure 4 (one MA server in the network) with Figure 5 (four MA servers in the network), we can see that if there are more MA servers deployed in the network, then more MAs will be dispatched to the network and fewer nodes will be left without receiving MAs. However, there will also be more nodes in the network to have received more than one MA.

The simulation results for randomly-deployed MA servers are plotted in Figure 6. From Figure 6 and Figure 5, we can see that when four MA servers are randomly deployed in the network, the number of nodes without MAs are generally more and the number of nodes with multiple MAs are less than when the MA servers deployed at the center of each network quadrant. The reason for this is that when MA servers are randomly deployed, there might be multiple MA servers in one quadrant and no MA server in another quadrant. Therefore, uneven distribution of MA servers will cause the MA distribution to be less effective.

From a comparison of the MA distribution results of having four randomly-deployed MA servers (as shown in Figure 6 (A) and (B)) against five randomly-deployed MA servers (as shown in Figure 6 (C) and (D)), we can see that the result is not much different. This is because there are already enough MAs dispatched to the network, and therefore dispatching more MAs or deploying more MA servers in the network simply will not help with the MA distribution.

VII. Related Work

There have been several different proposals for the design of IDSs for MANETs. We list them below and compare some of them with our proposed approach.

Zhang and Lee [9] proposed a distributed and cooperative intrusion detection model for MANETs. In their model, every node in the network runs an IDS agent, and performs data collection and intrusion detection locally, with cooperative detection and global response triggered whenever a node reports detection of an anomaly. The intrusion detection architecture is based on statistical anomaly detection techniques [9]. The internal structure of the IDS agent is divided into six models, as shown in Figure 7.

Sergio Marti *et al.* [10] discussed two techniques, watchdog and pathrater, that improve throughput in MANETs in the

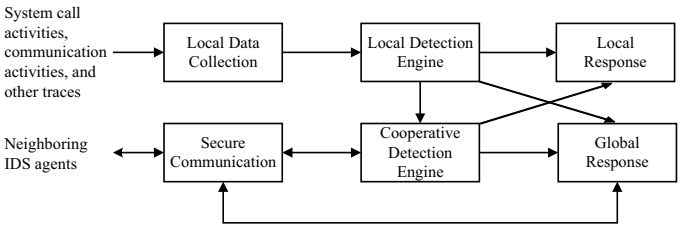

Fig. 7. The intrusion detection system for MANETs proposed in [9].

presence of compromised nodes that agree to forward packets but fail to do so. Watchdogs will identify misbehaving nodes, and pathraters will aid routing protocols to avoid misbehaving nodes. However, the watchdog technique might not detect misbehaving nodes under certain conditions.

Several researchers proposed the use of MAs for intrusion detection in MANETs. Local Intrusion Detection System (LIDS) proposed by Albers *et al.* [11] utilized MAs on each node in a MANET. LIDSs on different nodes collaborate by using security data to obtain complementary information from collaborating hosts, and intrusion alerts to inform others of a locally-detected intrusion. The LIDS agent could use either anomaly or misuse detection. Our approach differs from LIDS in that LIDS is a cooperative approach and the decision is based on the data collected from collaborating nodes. Therefore, LIDS will not function well if compromised nodes broadcast false intrusion-related information.

An intrusion detection architecture based on a static stationary database has been proposed by Smith [12], where each node in a MANET has an IDS agent running on it. The IDS agents on each node work together via a cooperative intrusion detection algorithm to decide when and how the network is being attacked. The architecture is divided into two parts, the mobile IDS agents that work on each node and the stationary secure database that has the signatures of all known misuse attacks and normal activity patterns of each node. This approach used both anomaly and misuse detection. Nevertheless, with a centralized database, the mobile nodes need to move and physically connect to the database periodically to stay up-to-date with the intrusion information. Also, the stationary secure database can become a single-point-of-failure if it is compromised. Our approach does not require the mobile nodes to physically move and connect to the MA server for update and can have multiple MA servers deployed in the network to avoid a single centralized server as an attack target.

Kachirske and Guha [13] also proposed a distributed IDS for wireless ad hoc networks based on the MA technology. By efficiently merging audit data from multiple network sensors, they analyzed the entire network for intrusions and tried to thwart intrusion attempts. They suggested not to use every possible node as IDS, but those chosen by a special "clustered network monitoring node selection algorithm." Designated nodes decide on intrusion. To avoid malicious votes, a modified independent decision-making system is used. This system uses a state machine for each known node, which is updated with threat information gathered by the monitoring agents. When a certain level or threat is reached, a command is sent to the node in danger, requesting necessary actions. Our approach has local IDS on each node and differs from the IDS in [13] where the monitoring and decision agents are on different nodes, thus requiring more communication and coordination.

VIII. Conclusion

We have proposed a distributed MA-based application-layer IDS for MANETs. The IDS utilizes both anomaly and misuse detection to identify attacks in MANETs. Also, the MAs augment each node's intrusion detection capability in the network by updating attack signatures and normal application profiles, patching and installing programs, further analyzing and diagnosing each node, and verifying the integrity of the IDS agents on each node. We have completed the design of the IDS architecture and the overall network structure, and described the design of protocols using MAs for the IDS. Our evaluation has demonstrated trade-offs between different design parameters of MAs.

Acknowledgement

The work reported in this paper was supported in part by a grant from the AFOSR Asian Office of Aerospace Research & Development.

References

[1] W. Jansen, P. Mell, T. Karygiannis, and D. Marks, "Applying Mobile Agents to Intrusion Detection and Response," in *NIST Interim Report (IR) 6416*, October 1999.

[2] T. Park and K. G. Shin, "Soft Tamper-Proofing via Program Integrity Verification in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 3, pp. 297–309, May/June 2005.

[3] A. Seshadri, A. Perrig, L. van Doorn, and P. Khosla, "SWATT: SoftWare-based ATTestation for Embedded Devices," in *Proceedings of the IEEE Symposium on Security and Privacy*, May 2004.

[4] T. Sander and C. F. Tschudin, "Towards Mobile Cryptography," in *Proceedings of the IEEE Symposium on Security and Privacy*, May 1998.

[5] P. Papadimitratos and Z. J. Haas, "Secure Routing for Mobile Ad hoc Networks," in *Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, January 2002.

[6] L. Luo, R. Safavi-Naini, J. Baek, and W. Susilo, "Self-organised Group Key Management for Ad Hoc Networks," in *Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security (ASIACCS 2006)*, March 2006.

[7] B. Wu, J. Wu, and Y. Dongand, "An Efficient Group Key Management Scheme for Mobile Ad Hoc Networks," *International Journal of Security and Networks*, vol. 4, no. 1/2, pp. 125–134, 2009.

[8] J. R. Douceur, "The Sybil Attack," in *Proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS 2002)*, March 2002.

[9] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad Hoc Networks," in *Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom 2000)*, August 2000.

[10] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom 2000)*, August 2000.

[11] P. Albers, O. Camp, J.-M. Parcher, B. Jouga, L. Me, and R. Puttini, "Security in Ad Hoc Networks: a General Intrusion Detection Architecture Enhancing Trust Based Approaches," in *Proceedings of the International 1st Workshop on Wireless Information Systems (Wis 2002)*, April 2002.

[12] A. B. Smith, "An Examination of an Intrusion Detection Architecture for Wireless Ad Hoc Networks," in *Proceedings of the 5th National Colloqium for Information System Security Education*, May 2001.

[13] O. Kachirski and R. Guha, "Intrusion Detection Using Mobile Agents in Wireless Ad Hoc Networks," in *Proceedings of the IEEE Workshop on Knowledge Media Networking (KMN 2002)*, July 2002.