SECURE POSITION-AIDED AD HOC ROUTING

Name: Stephen H. Carter
Department: Computer Science Department
Major Professor: Alec Yasinsac
Degree: Master of Science
Term Degree Awarded: Fall, 2002

Position-aided routing protocols can offer a significant performance increase over traditional ad hoc routing protocols. These routing protocols use geographical information to make forwarding decisions, resulting in a significant reduction in the number of routing messages. However, current position-aided routing protocols were not designed for use in high-risk environments because position information is broadcasted in the clear, allowing anyone within range, including the enemy, to the exact location of each node. In this paper we study methods of protecting position information in ad hoc routing protocols and ways to use the position information to enhance performance and security of these routing protocols. We introduce Secure Position-Aided Ad hoc Routing (SPAAR), a group of protocols designed to protect and make use of position information to improve security, efficiency, and performance in ad hoc routing.

THE FLORIDA STATE UNIVERSITY

COLLEGE OF ARTS & SCIENCES

SECURE POSITION-AIDED AD HOC ROUTING

By

STEPHEN H. CARTER

A thesis submitted to the
Computer Science Department
in partial fulfillment of the
requirements for the degree of
Master of Science

Degree Awarded:
Fall Semester, 2002

The members of the Committee approve the thesis of Stephen H. Carter defended on November 8, 2002.

Alec Yasinsac
Professor Directing Thesis

Kohout
Committee Member

Ladislav
Committee Member

Approved:

Sudhir Aggarwal, Chair
Department of Computer Science

This is dedicated to a number of people without whom I may have never been able to write this thesis. To my mother and father, Susan and Bill Carter, who have always been my role models and who have given me the opportunity to be what I am today. I could not have made it this far without your unconditional love and support. You have always set an excellent example for me to follow and I thank God I have been blessed with the best parents a person could ask for. To Kelsie Willett, my best friend, who made the last seven years the best years of my life. I do not think I could have survived in Tallahassee without your love, support, patience and good cooking. To my family and friends, especially my old friends in Jacksonville, I could have never done this without you. I have not been able to see or talk with many of you nearly as much as I would have liked in the past four years. Nevertheless, you have helped me more during this time than you can imagine. Your faith and confidence in me gave me motivation to work hard and determination to never give up.

ACKNOWLEDGEMENTS

I am very grateful to my Major Professor, Dr. Alec Yasinsac, for his guidance and support over the past year. He has shown me the importance of research and his enthusiasm for his work has been a great motivation for me. I would also like to thank my committee members, Dr. Mike Burmester and Dr. Ladislav Kohout, for their insightful comments and suggestions regarding my thesis. A special thanks to John Marshall (JTS) for all of his help with my thesis over the past few months.

I would like to thank Dr. James O'Brien, Jordan Yao and everyone at the Center for Oceanic and Atmospheric Prediction Studies (COAPS), for giving me the opportunity to gain valuable work experience in such a unique, challenging and friendly environment, while attending graduate school. Most of all, I thank God for everything and everyone I have in my extraordinary life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Position-aided routing protocols can offer a significant performance increase over traditional ad hoc routing protocols. These routing protocols use geographical information to make forwarding decisions, resulting in a significant reduction in the number of routing messages. However, current position-aided routing protocols were not designed for use in high-risk environments because position information is broadcasted in the clear, allowing anyone within range, including the enemy, to the exact location of each node. In this paper we study methods of protecting position information in ad hoc routing protocols and ways to use the position information to enhance performance and security of these routing protocols. We introduce Secure Position-Aided Ad hoc Routing (SPAAR), a group of protocols designed to protect and make use of position information to improve security, efficiency, and performance in ad hoc routing.

CHAPTER 1

INTRODUCTION

Mobile Ad hoc Networks (MANETs) are wireless networks lacking a fixed infrastructure, in which the nodes are free to move about arbitrarily resulting in a highly dynamic network topology [1]. The nodes in a MANET may change position at any time or adjust their transmission and reception parameters, causing links to be broken and re-established. Nodes are dependent on each other to keep the network connected, as each node generally functions as a router [2, 1]. These salient characteristics of MANET make traditional fixed-network routing protocols inadequate.

Ad hoc network research has resulted in a number of routing protocols suitable for use in MANETs [3]. Most current research in MANET routing is focused on topology-based protocols. Topology-based routing protocols use the information about links that exist in the network to perform packet forwarding and are generally classified as either table-driven or on-demand. Table-driven protocols, such as Destination Sequenced Distance Vector Routing (DSDV), require that each node maintain information about available paths in the network, even if they are not currently used. Nodes attempt to maintain consistent up-to-date routing information from each node to every other node in the network [3, 4]. The primary drawback of this approach is the high amount of overhead and storage involved in maintaining these paths.

On-demand protocols, such as Ad-hoc On-Demand Distance Vector Routing (AODV) and Dynamic Source Routing (DSR), only maintain routes that are cur-

rently in use. As a result, on-demand protocols usually require very little overhead. Nevertheless, on-demand protocols have some limitations. Nodes must initiate a route discovery process when they wish to acquire a path to an unknown destination. This process means a delay for the first packet on a new path. The performance of on-demand protocols usually decreases with the increase of node mobility, due to the overhead involved in the route discovery process.

Research has shown that position-based routing protocols are a good alternative to on-demand protocols in many cases [4, 5, 6]. Position-based routing protocols use nodes' geographical positions to make routing decisions, resulting in an improved efficiency and performance. These protocols require that a node is able to obtain its own geographical position and the geographical position of the destination. Generally this information is obtained via Global Positioning System (GPS) and location services.

One primary application of MANET is in military use including tactical operations. In these environments, security is often the primary concern. Secure routing protocols are needed to protect routing messages against malicious nodes and attacks that one could expect in hostile environments.

The routing protocol used in a network sets an upper bound on the security of the network. If routing can be misdirected, the entire network can be paralyzed [7]. Most traditional topology-based MANET protocols were designed with reliability and performance in mind. Unfortunately these protocols were not designed to be secure and do not defend against malicious attacks. AODV and DSR, two protocols under consideration for standardization by the IETF MANET Working Group, are both vulnerable to a number of attacks, including impersonation, modification, and fabrication [4]. Position-based MANET routing protocols [8, 9, 5] are also vulnerable to such attacks, as they focus on improving performance while disregarding security issues. In addition, these protocols lack cryptographic techniques to protect location information exchanged between nodes, revealing the exact location of nodes to anyone

within range. In a high-risk environment, this is unacceptable. Cryptographic techniques must be employed to protect position information in these protocols, if they are to be used in a high-risk MANET.

If position information can be safely protected, it can be used to improve the efficiency and security of MANET routing. We introduce SPAAR, a group of protocols designed to provide secure on-demand position aided routing [3] in MANET. SPAAR protects position information via public key cryptography and uses the position information to make forwarding decisions in the route discovery process, resulting in less routing overhead. The protected position information is also used to determine one-hop neighbors with whom a bi-directional communication is possible. By verifying one-hop neighbors, and only accepting routing messages from these neighbors, SPAAR defends against certain attacks that other secure routing protocols are vulnerable to [10, 11].

In the following chapter we discuss previous work relating to secure routing protocols and position-aided routing protocols. We describe an attack on the Secure Routing Protocol (SRP) that is a large part of the motivation for SPAAR. In chapter three, we describe the target environment for SPAAR and give a detailed description of the protocols that SPAAR is comprised of. In chapter four, we discuss how SPAAR satisfies the security requirements for routing in a high-risk environment and give attack scenarios to illustrate. We consider attacks by non-colluding malicious nodes and colluding malicious nodes. We follow up in Chapter five with conclusions and recommendations for future work.

CHAPTER 2

RELATED WORK

SPAAR is a group of protocols that provides secure position-aided ad hoc routing, therefore related work includes secure ad hoc routing protocols and position-aided ad hoc routing protocols. In this chapter we discuss recent research in secure routing and position-aided routing protocols for ad hoc networks.

2.1 Secure Routing Protocols

Ad hoc network research has produced numerous routing protocols, some of which are under consideration by the IETF for standardization [16, 15]. Most of these routing protocols fall short of providing any significant level of security, so their use is limited. With the realization of the possibilities for use of MANETs in military applications, the need for secure routing protocols has grown.

Substantial progress has been made in the design of secure routing protocols for MANETs [12, 13, 14]. Though not without flaws, from a security standpoint these protocols are drastic improvements over traditional MANET routing protocols [15, 16].

2.1.1 The Secure Routing Protocol

Papadimitratos and Haas [13] propose the Secure Routing Protocol (SRP) as a solution for securing MANETs. SRP requires a security association between the source and destination nodes and the authors assert that SRP guarantees the node initiating a route discovery will be able to identify and discard replies providing false

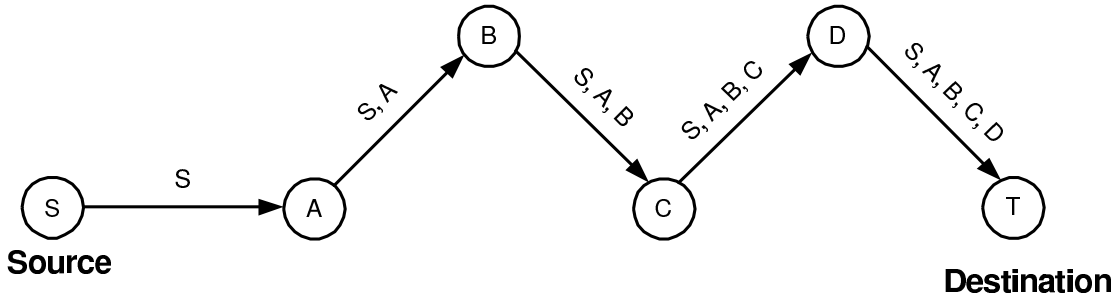

Figure 2.1. The SRP Route Discovery Process

topological information, or avoid receiving them altogether. SRP is implemented as an extension to a base reactive protocol, such as AODV. The authors present a number of possible attack scenarios, and describe how their protocol thwarts these attacks.

SRP makes use of a route field in route request (RREQ) and route reply (RREP) packets. Each intermediate node appends its identifier to the route field as a routing packet propagates from the source to the destination. An illustration of how the route field is maintained appears in Figure 2.1.

Marshall [10] points out a weakness in SRP and presents an attack. The premise of Marshall's attack is that a malicious node M may forward a RREQ without appending its address to the field of the SRP header, effectively making itself invisible in the path returned to the source. As Figure 2.2 illustrates, the result is the source node erroneously believes that a path exists to a destination that is not dependent on M.

Initially, one might question the significance of this attack. While Marshall [10] describes the attack in detail, little is mentioned of the possible effects. We feel that this attack is much more significant than it may first appear. One consequence of this attack could be fooling S into using a path that appears ideal, but may not have appeared ideal if the malicious node (or nodes) was visible. For example, suppose SRP was being used as an extension to a shortest path routing algorithm

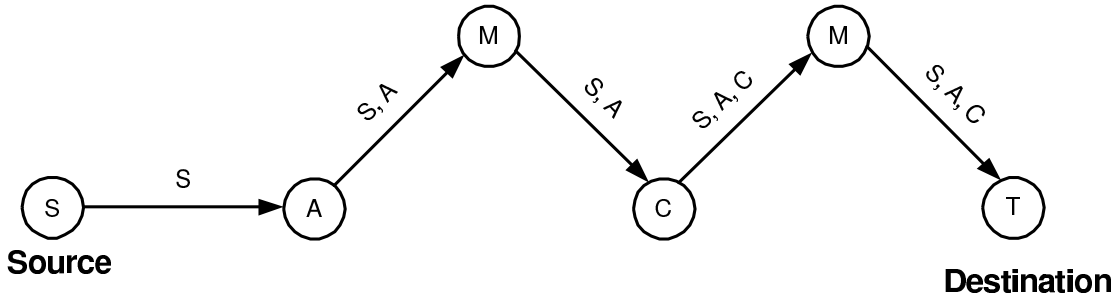

Figure 2.2. Attack on SRP Route Discovery Process

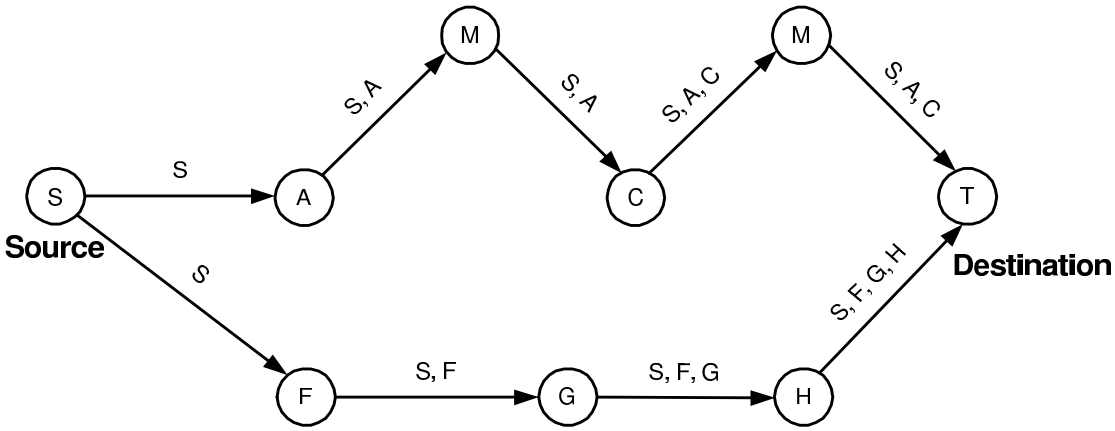

Figure 2.3. Attack Scenario Involving Non-colluding Malicious Nodes

that measured path length as the number of hops. In this case, S may decide to use a path that appears to be the shortest, but in actuality is not because one or more hops are invisible in this path due to malicious nodes. This scenario is illustrated in Figure 2.3.

In Figure 2.3, the true shortest path (S, F, G, H, T) only requires four hops, however the source node will not choose this path because it believes in the false path of three hops (S, A, C, T). If S does choose this path, the malicious nodes could then negatively impact network performance by intentionally delaying packing or dropping packets. Despite the poor performance, the base protocol may continue to choose this path as ideal since it appears to be the shortest route.

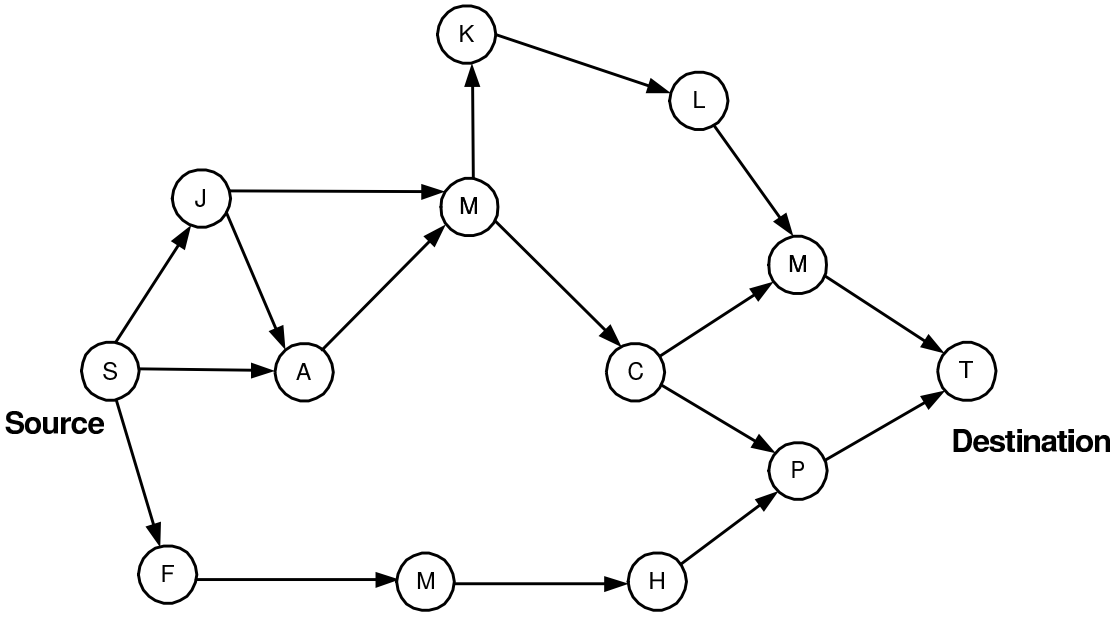

Figure 2.4. Attack Scenario Involving Colluding Malicious Nodes

While malicious nodes need not collude to execute the invisible node attack, multiple malicious nodes may collude to disrupt or temporarily disable a MANET using SRP. Suppose a number of colluding malicious nodes strategically place themselves in positions where they are essential links in a large number of routes, as depicted in Figure 2.4. If at a certain pre-determined time, all malicious nodes stop forwarding packets, the network would be crippled due to the sudden number of broken links.

SRP may be sufficient for use in certain environments. However, it is not sufficient for the high-risk environment targeted in this paper because it does not satisfy security requirements four, five, and seven. Routes can be redirected from the shortest path by a malicious unauthorized node on the network, as we have illustrated. SRP does not require nodes to be authorized before participating in the protocol. In addition, the routing messages are not authenticated. SRP takes no measures to protect topology information from malicous nodes. Route records are

passed in the clear, exposing the path a packet has traveled thus far. Revealing this topological information is unacceptable in a high-risk environment.

2.1.2 Other Secure Routing Protocols

In [14] the Security-Aware Ad-hoc Routing Protocol (SAR) is introduced. Like SRP, this protocol is an extension (or augmentation) to existing on-demand ad hoc routing protocols. However, SAR takes a different approach to secure routing. Rather than proposing a specific solution to ad hoc routing, the authors present a generalized framework that allows the user to specify the security level that should be used in the routing protocol.

In SAR nodes are assigned trust values and data is routed only through trusted nodes. The source sends a RREQ with embedded security attributes and trust levels are defined by the user. Only those nodes that satisfy the required level of security can participate in the routing protocol. Nodes that do not meet the requested security requirements must drop the RREQ. If a route satisfying the requested security attributes does not exist, the protocol initiator can choose to send another RREQ with modified security attributes to find a route with different security guarantees.

SAR is flexible in the sense that it may be used in many different ad hoc environments. However, being only a framework, it is incomplete in the sense that the authors do not give enough details to implement SAR for use in a real world environment. SAR does not discuss how a node is assigned a trust value or how applications determine the level of trust needed. In addition to these issues that must be resolved before SAR could be implemented, SAR is also vulnerable to the SRP attack described in [10]. We feel that the framework provided by SAR is not sufficient to meet the security requirements of the high-risk environment that we target in this paper.

In [12] the authors present a secure routing protocol for ad hoc networks called ARAN (Authenticated Routing for Ad hoc Networks). ARAN uses public

key cryptographic certificates for authentication and non-repudiation to satisfy the security requirements for the environment described by the author as the managed-open environment. Privacy is not guaranteed resulting in the exposure of network topology. The author acknowledges this fact and states that ARAN is not suitable for use in the high-risk managed-hostile environment.

2.2 Position-Aided Routing Protocols

In topology-based protocols, route discovery entails flooding a RREQ packet to all neighbors. In many cases this technique is wasteful. The entire MANET may be involved in a route discovery when only a small percentage of nodes, those closer to the destination than the source, should be involved. In response to this observation, MANET research has produced a number of position-based routing protocols that offer a significant performance increase over topology-based protocols in certain environments [8, 4, 5, 9]. Although each of these protocols employs different techniques, the basic goal is the same. Only nodes making forward progress toward the destination should be involved in the route discovery process, resulting in a significant decrease in routing overhead.

2.2.1 Location-Aided Routing

In [5] the authors present an approach to routing that utilizes location information to decrease the overhead of route discovery called Location Aided Routing (LAR). Two routing schemes are introduced that use position information to make forwarding decisions. LAR reduces the search space for a desired route, resulting in fewer route discovery messages.

In LAR scheme 1, when a source node S wants to initiate a route discovery for a destination D it will first compute D's *expected zone*. The expected zone is defined as the region that node S expects to contain node D at a particular time t1. Node

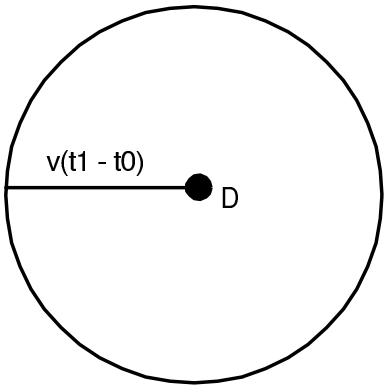

Figure 2.5. LAR Expected Zone

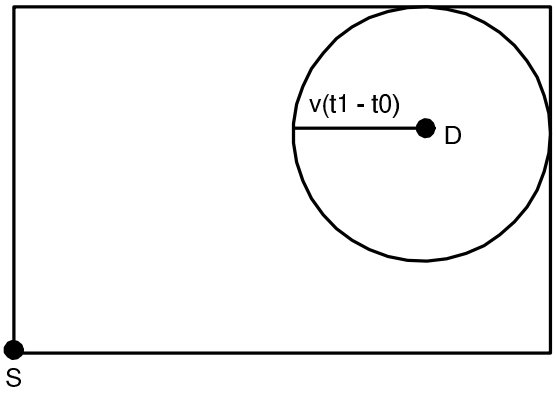

Figure 2.6. LAR Request Zone

S may determine the expected zone based on the knowledge that node D was at location L at time t0 and that node D travels with average speed v (see Figure 2.5).

Next, node S defines a *request zone* for the RREQ. The request zone is defined to be the smallest rectangle that includes the current location of S and the expected zone, such that the sides of the rectangle are parallel to the X and Y axes (see Figure 2.6). S determines the coordinates of the four corners of the request zone rectangle and includes them in the RREQ. An intermediate node will forward a RREQ only if it lies inside the request zone specified in the RREQ.

When node D receives the RREQ, it replies by sending a RREP that contains its current location and speed (or average speed over a recent time interval). Node S uses this information for future route discoveries.

In LAR scheme 2, the source node S calculates the distance to destination D, denoted DISTs, and includes this information along with the D's coordinates in the RREQ. Upon receiving a RREQ, an intermediate node I will compute its own distance to D (DISTi) and will forward the RREQ only if it is at most d farther from the destination that the previous node (DISTs + d $\geq$ DISTi, for some parameter d).

2.2.2 GPSR: Greedy Perimeter Stateless Routing

In [8] Karp and Kung present Greedy Perimeter Stateless Routing (GPSR), a routing protocol for wireless datagram networks that uses position information of nodes to make forwarding decisions. GPSR makes greedy forwarding decisions using position information about a node's immediate neighbors and the destination. Specifically, a node only forwards a RREQ to the neighbor geographically closer to the destination. In certain regions where greedy forwarding fails GPSR recovers by routing around the perimeter of the region.

GPSR assumes that all nodes have an identical circular radio range r. The network is viewed as a graph where each node is a vertex and an edge (n,m) exists between nodes n and m if the distance between n and m is less than or equal to r. The graph is converted to a planar graph, where no two edges cross. The authors describe methods for a node to remove certain links with one-hop neighbors to create a Relative Neighborhood Graph (RNG) or a Gabriel Graph (GG), both of which are planar graphs.

When greedy forwarding fails, the packet enters *perimeter mode*. GPSR forwards perimeter mode packets using a simple planar graph traversal (see Figure 2.7). If a packet enters perimeter mode at node x bound for node D, the packet is forwarded on progressively closer faces of the planar graph, each of which is crossed by the line xD.

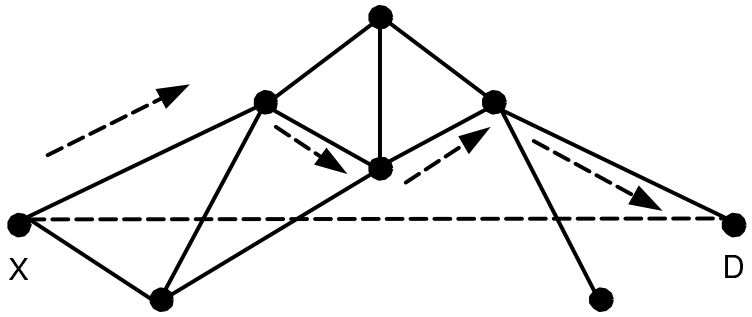

Figure 2.7. Perimeter Forwarding

A planar graph consists of two types of faces. Closed polygonal regions bounded by the graph's edges are called interior faces. The one unbounded face outside the outer boundary of the graph is known as the exterior face. On each face, the traversal uses the right-hand rule to reach an edge that crosses line xD. At that edge, the traversal moves to the adjacent face crossed by xD. This process is repeated until the destination is reached or the protocol can re-enter greedy forwarding mode.

CHAPTER 3

SECURE POSITION-AIDED AD HOC ROUTING (SPAAR)

In this chapter we introduce SPAAR. We will begin by discussing some important features of SPAAR and describing the high-risk environment that SPAAR is designed for. We introduce seven security requirements for secure routing in such a high-risk environment. Next we describe the SPAAR neighbor table, the protocol for adding a neighbor, and neighbor table maintenance protocols. We conclude this chapter with a description of the route discovery protocol and route table maintenance protocols.

3.1 SPAAR Features

SPAAR uses position information to improve performance and security, while keeping position information protected from unauthorized nodes. For MANET routing protocols to achieve a high level of security, it is imperative that a node only accepts routing messages from verified one-hop neighbors. In SPAAR, a node can verify its one-hop neighbors before including them in the routing protocol. This is made possible by the use of geographical location information. SPAAR requires that each device have some means of determining its own location. GPS receivers have become relatively inexpensive and lightweight. Therefore, we feel it is reasonable to assume that all devices in our network could be equipped with a GPS receiver. In addition, recent advances in GPS security make it more practical for use in a high-risk environment [17]. In the case that a node is unable to determine its location either

due to lack of a GPS receiver or terrain obstacles, a node may use a location proxy as described in [6].

In SPAAR, the source node must also know the geographic location (or an approximation) of the destination. This may be calculated from the most recent location and most recent velocity information stored in the source node's route table. However, if this is the source node's first attempt at communication with a particular destination, the source has no way of calculating the destination's position. In this situation, a selective flooding algorithm is used to reach the destination and receive its position information. While a location service [18, 5] is not assumed, the use of such a service would significantly reduce the overhead involved in SPAAR.

SPAAR makes use of a trusted certificate server. An alternate implementation of SPAAR that takes advantage of the opportunity for the exchange of security parameters prior to node deployment may be possible under certain conditions. This implementation would not require a trusted certificate server. In the target environment, a tactical plan of some sort usually exists. In many cases it is possible to designate the set of nodes that a particular node will communicate with as a one-hop neighbor. Depending on the size of this set, a node could store these nodes' public keys in non-volatile memory. Although this solution does not scale, it may be applicable in the target environment, and when applied it eliminates the need for a trusted certificate server in our protocol.

With SPAAR, one could use any one of the different geographic forwarding techniques to make forwarding decisions, with little modification to the SPAAR protocol. For simplicity's sake, we chose to use LAR scheme 2 with d = 0 [5].

3.2 SPAAR Environment

Due to the numerous applications of ad hoc networks, different ad hoc routing protocols must be designed for and tailored to specific environments. SPAAR was designed for use in a high-risk tactical MANET.

A routing protocol may be considered secure if it meets the security requirements for its environment of use. In [12] the authors classify ad hoc networks into three environments: open, managed-open, and managed-hostile. Each environment differs greatly in its security needs and the opportunity for pre-deployment coordination. The authors describe a secure routing protocol designed for the managed-open environment, where security is a concern, though not the primary concern. SPAAR targets an environment similar to the managed-hostile environment. The goal of SPAAR is to secure routing in such a high-risk environment by satisfying the set of security requirements listed in Table 3.1, which are an adaptation of the security requirements of the managed-hostile environment described in [12].

Table 3.1. Security Requirements

$SR1$	Fabricated routing messages cannot be injected into network by malicious nodes
$SR2$	Routing messages cannot be altered in transit by malicious nodes
$SR3$	Routing loops cannot be formed by malicious nodes
$SR4$	Routes cannot be redirected from the shortest (or ideal) path by malicious nodes
$SR5$	Unauthorized nodes should be excluded from route computation and discovery
$SR6$	Network topology must not be exposed to malicious nodes by routing messages
$SR7$	Nodes must not store inaccurate routing information as a result of malicious node activity

The managed-hostile environment is described as a MANET formed by military nodes in a battle environment or emergency response crews in a disaster area. In this type of environment, security is essential and the protection of node location is often necessary. Nodes are generally deployed from a common source and the opportunity for the pre-deployed exchange of security parameters often exists. Sensitive information is passed between nodes, and malicious nodes are a constant threat.

It is important to distinguish *malicious nodes* from *compromised nodes*. SPAAR is designed to defend against malicious nodes. For the purpose of this paper, we define a malicious node to be an unauthorized node attempting to disrupt or attack the network. Adversaries deploy malicious nodes to engage in malicious activity such as eavesdropping, message replay, message distortion, and impersonation. SPAAR makes use of encryption to thwart such attacks.

We define a compromised node to be an authorized node deployed by a known source but has been overtaken by an adversary. Compromised nodes can produce valid signatures and possess valid certificates. A compromised node may or may not engage in malicious activity or misbehave. As a result, detection of compromised nodes can be very difficult. In many cases it is difficult to distinguish malicious activity by a compromised node from legitimate node activity.

SPAAR protects a MANET from attacks by malicious nodes. We recognize the importance of defending against compromised nodes, but defending against attacks by malicious nodes is our first priority. The goal of SPAAR is to prevent attacks from malicious nodes while minimizing the potential for damage from attacks by compromised nodes.

While a SPAAR-protected network will not be safe from all malicious attacks by compromised nodes, intrusion detection systems (IDS) can help to identify compromised nodes and mitigate routing misbehavior. In [19] Zhang and Lee introduce an intrusion detection system for ad hoc networks. In their approach, every node participates in intrusion detection and response. Each node is responsible for detecting signs of intrusion locally and independently. Neighboring nodes can collaborate to investigate in a broader range if necessary. Individual agents run independently on every node monitoring local activity, collectively forming an IDS to defend wireless ad hoc networks.

Methods of detecting and mitigating routing misbehavior in MANETs are discussed in [20]. The authors present two routing protocol extensions to mitigate

routing misbehavior: the watchdog and the pathrater. The watchdog identifies misbehaving nodes while the pathrater uses this knowledge of misbehaving nodes to choose the network path most likely to deliver the packets. Watchdog and pathrater increase overall routing overhead, however this is offset by the increase in network throughput in the presence of misbehaving nodes.

3.3 SPAAR Setup

SPAAR does not require a pre-existing online key management system in the MANET. Knowledge of the public keys of other network nodes, or a service that provides the public keys of all nodes on the network, is not required. SPAAR does require that each node have access to a trusted certificate server before it can participate in the routing protocol. Because the targeted environment generally affords some amount of node preparation prior to deployment, we assume that nodes have access to such a certificated server before entering the MANET. To participate in SPAAR, each node requires a public/private key pair, a certificate binding its identity to its public key (signed by a trusted certificate server), and the public key of the trusted certificate server.

All nodes are deployed with the private part of a public/private key pair. Prior to deployment, each node will request a certificate from a trusted certificate server T. The certificate binds a node's identity with its public key and is signed by T. The certificate is time stamped and has an expiration time. Each node will possess T's public key so it can decrypt certificates of other nodes. This allows a node N1 to inform another node N2 of its public key, assuming node N2 was deployed correctly with T's public key to decrypt certificates.

Certificate = [identity, public key, time, expiration] T_k-

3.4 The Neighbor Table

With SPAAR, each node maintains a neighbor table that contains the identity and position information of each verified neighbor, along with the cryptographic keys required for secure communication with each neighbor (See Table 3.2). A node only accepts routing messages from a node in its neighbor table.

Table 3.2. Neighbor Table

ID	Neighbor's identification
PK	Neighbor's public key verified from its certificate
GDK	Neighbor's group decryption key
MRL	Neighbor's most recent location (lat/long coordinates)
$LUSN$	Neighbor's location update sequence number
TR	Neighbor's transmission range

3.4.1 Adding Nodes to the Neighbor Table

Adding nodes to the neighbor table is a three-step process that is illustrated in Figure 3.2. In step one, a node N broadcasts a HELLO message with its certificate CERT_N. Any nodes within range of N, wishing to be recognized as a neighbor, decrypt N's certificate to verify N's public key and create an entry for N in the neighbor table where N's public key will be stored.

In step two, nodes respond to N with a hello reply (HELLO_REP) that includes their certificate, MRL, and TR signed with their public key and encrypted under N's public key. Upon receiving a HELLO_REP from a neighbor node X1, N will verify that X1 is truly a one-hop neighbor with the method in Figure 3.1.

If N has verified the node as a one-hop neighbor, in step three N will store the node's public key, most recent location, and transmission range in N's neighbor table. If this is the first neighbor to be added to the neighbor table, N will generate a public/private key pair, which we call a neighbor group key pair. The private part of N's neighbor group key pair will be called N's group encryption key and denoted

```
verified_neighbor = FALSE

if message == HELLO_REP

        distance = compute_distance (N's coordinates, X1's coordinates)

                if distance < N's transmission range &&
                   distance < X1's transmission range
                then
                   verified_neighbor = TRUE

end if
```

Figure 3.1. Method for Verification of One-hop Neighbors

GEK_N. The public part of node N's neighbor group key pair will be called N's group decryption key, denoted GDK_N. N distributes GDK_N to each of his neighbors once they have been verified as one-hop neighbors. The GDK is signed with N's private key to provide authentication and encrypted under the neighbor's public key for privacy. Upon receiving the GDK_N, N's neighbors store it in their neighbor table.

At this point X has the capability to accept routing packets from N. However, X will not do so until it has verified N as a neighbor. This will occur after X broadcasts a HELLO message and the above steps are executed. This table state will last, at most, the time between HELLO message broadcasts of X.

3.4.2 Neighbor Table Maintenance

Each node periodically (every n seconds) broadcasts a table update message to inform the neighbors of its new MRL, TR, and LUSN. Table update messages are encrypted with a node's group encryption key. Neighbors of N decrypt the table update message, analyze the new position information to verify that the neighbor is still a one-hop neighbor, and update their neighbor table with the new position information.

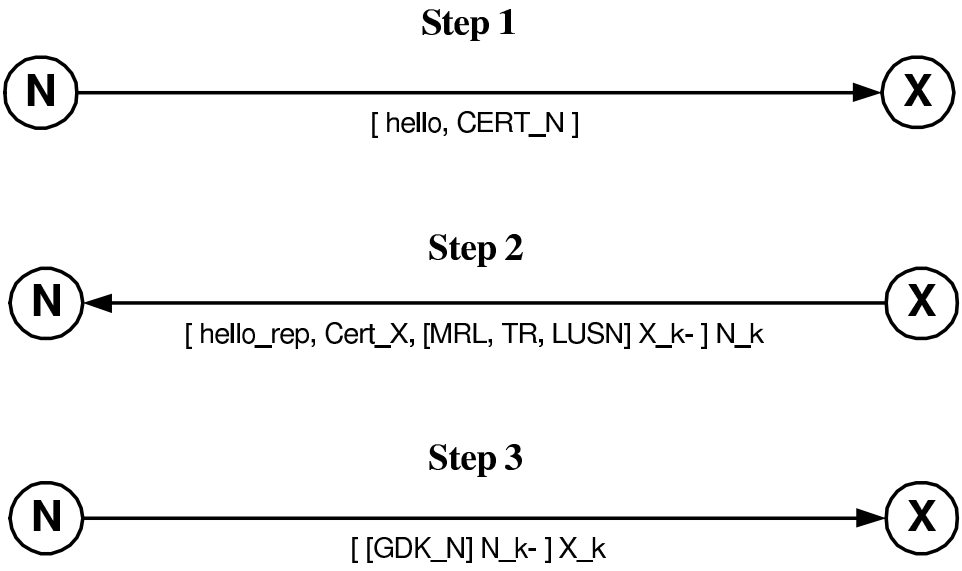

Figure 3.2. Adding A Node To The Neighbor Table

The location update sequence number, LUSN, is a time-stamped sequence number that is incremented each time N broadcasts a table update message or constructs a RREP containing its position information. Representing the freshness of location information, the LUSN prevents replay attacks of table update messages. A node uses the LUSN in the RREQ to inform its neighbors of the freshness of the coordinates it possesses for the destination.

When a table update message is received, the LUSN is time-stamped allowing the node to determine how much time has passed since it has received a table update from its neighbors. It should be noted that the LUSN time stamp is not the exact time of the MRL coordinates for a destination. The MRL coordinates are from time t = (LUSN time stamp - propagation delay of the message that included the LUSN). After a timeout period has elapsed without a table update from a neighbor, the link to that neighbor is assumed to be broken and the neighbor is deleted from the neighbor table.

The interval at which a node broadcasts a table update depends on its rate of mobility. A node with a high mobility rate will broadcast table update messages

more frequently in an effort to keep its neighbors up-to-date. To offset the overhead involved with such a proactive approach, table update messages are piggybacked on all routing messages encrypted with a node's neighbor group key (RREQ and location request messages).

3.4.3 Hello Messages

All nodes broadcast periodic HELLO messages allowing for new neighbors to be added to the neighbor table. The HELLO message contains the sender's public key certificate. A node receiving a HELLO message from N checks to see if N is already in its neighbor table. If so, the node then checks to see if the GDK field has a value. If the node has a value for node N's GDK field, it is already in N's neighbor group and will ignore the HELLO message. If a node does not have N in its neighbor table, or it has no value for the node's GDK field in the neighbor table, it will send a HELLO_REP message as previously described. As with table update messages, the interval between HELLO messages is dependent upon node mobility.

3.5 Route Discovery and Route Maintenance

In SPAAR, a source node initiates the route discovery process by broadcasting a RREQ. Upon receiving the RREQ, the destination node responds with a RREP. The route discovery process is decribed in detail in the following sections.

3.5.1 Route Requests (RREQ)

A Node N begins the route discovery process by calculating an estimation of the destination's current position (velocity x age of position coordinates). Next, N broadcasts a RREQ containing the RREQ_SN (see Table 3.2), the destination's identifier, N's distance to D, the destination's MRL, and the destination's LUSN, all encrypted with its group encryption key (see Figure 3.3). The RREQ_SN is

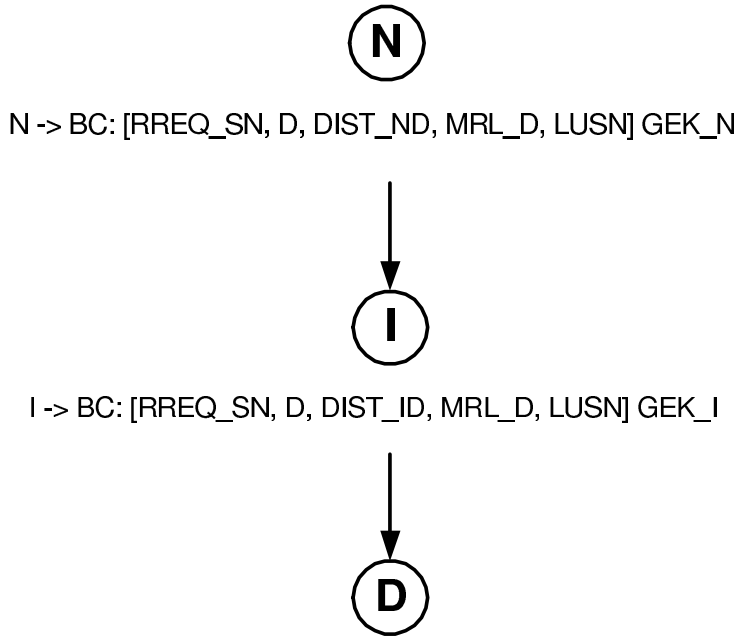

Figure 3.3. Route Request (RREQ) Propagation

incremented each time a node transmits a RREQ. It is used to prevent replay attacks of RREP and RERR messages.

Recipients of the RREQ, that are neighbors of N, decrypt it with N's group decryption key. A successful decryption of a RREQ implies that the sender of the RREQ is a one-hop neighbor. As LAR scheme 2 specifies, an intermediate node checks to see if it is closer to destination D. If an intermediate node has the destination's coordinates with a more recent LUSN, it uses those coordinates for the comparison instead of the coordinates contained in the RREQ.

If the intermediate node is not closer to the destination, the RREQ is dropped. If either is closer, the node re-broadcasts the RREQ with its identifier and distance to S, encrypted with its group encryption key. If the intermediate node has the destination's coordinates with a more recent LUSN, those coordinates replace the older coordinates in the RREQ. Intermediate nodes record,in their route table (see Table 3.3), the address of the neighbor from which they received the RREQ, thereby establishing a reverse path. This process is repeated until the destination is reached.

3.5.2 The Route Table

Table 3.3. Route Table

$RREQ\_SN$	Route request sequence number used to identify a RREQ
S/DID	The source and destination ID's
$REVERSE$	The next hop in the reverse path to the source
$FORWARD$	The next hop in the forward path to the destination
MRL	Destination's most recent location
TR	Destination's transmission range
$LUSN$	Destination's location update sequence number
VEL	Destination's velocity
A/I	The active/inactive flag

Each node maintains a route table containing the fields shown in Table 3.3. An entry in the route table is created when a RREQ is received or a node initiates a route discovery. The RREQ_SN from the route request is stored to prevent RERR replay attacks as discussed in chapter 4. The source and destination addresses associated with the route request are also stored. The reverse field is the address of the node from which the RREQ was received and the forward field is the address of the node from which the corresponding RREP was received. The location information for the destination is stored in the MRL, TR, and LUSN fields. Each route in the route table is initially marked active, however a route may be deactivated for a number of reasons discussed in Section 3.5.5.

3.5.3 Route Replies (RREP)

Upon receiving a RREQ, the destination constructs a RREP containing the RREQ_SN, its MRL, its velocity, and a LUSN. The destination's certificate is also included enabling any node to verify the destination's signature on the contents of the RREP. The destination signs the RREP with its private key and encrypts it with the public key of the neighbor from which it received the RREQ. The RREP propagates along the reverse path of the RREQ, being verified at each hop (see Figure 3.4).

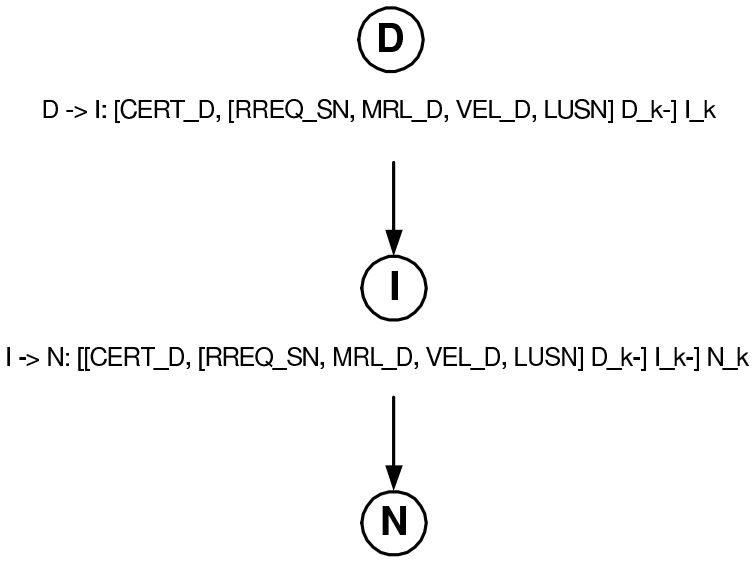

Figure 3.4. Route Reply (RREP) Propagation

Intermediate nodes, upon receiving a RREP, decrypt it with their private key and verify the signature with the public key of the neighbor node they received it from. Next, the contents of the RREQ are decrypted with the public key of the destination. If the decryption is successful, a forward entry is then added to the intermediate node's route table that points to the node from which the RREP was received. An unsuccessful decryption implies that the contents of the RREP have been tampered with, and the RREP is discarded. Intermediate nodes sign the RREP and encrypt it with the public key of the next node in the reverse path. The RREP is then forwarded to the next node in the reverse path.

An intermediate node may receive many RREPs in response to one RREQ. The first RREP received is the one that will be used, however intermediate nodes create entries in their routing tables for the first three RREPs they receive from different nodes for a given RREQ. The storage of alternate routes (redundant paths) helps a node recover from broken links and helps to prevent damage caused by compromised nodes as discussed in chapter 4.

Following the successful receipt and authentication of a RREP, the source node verifies that the RREQ_SN matches the RREQ_SN from the initial RREQ. This is done to prevent replay of RREPs by malicious nodes. The source node then creates a new entry in the route table. The source node time-stamps the LUSN so it can determine how much time has passed since the last update. As with intermediate nodes, the source node will use the route from the first RREP it receives and create entries in its routing table for the first three RREPs it receives from different nodes for a given RREQ. In the case that a source node does not receive a RREP in response to a RREQ, flooding must be used.

3.5.4 Location Request Messages

There will be cases when a node has no previous location information for a destination to include in the RREQ. In this case, a node broadcasts a location request message to its neighbors in an attempt to discover the location of the destination.

$$N \rightarrow BC: [LOC\_REQ, D]\ GEK\_N$$

Any neighbors that possess the location coordinates for the destination will respond to S with a signed location reply, encrypted with N's public key.

$$Neighbor \rightarrow N: [[LOC\_REP, D, MRL\_D, VELOCITY\_D, LUSN, age]\ Neighbor\_k\text{-}]\ N\_k$$

SPAAR does not assume clock synchronization between nodes, therefore the local timestamp on a LUSN is irrelevant to another node. For this reason, when a node sends a location reply, it includes the age of the position information. The age is equal to the time that has passed since the LUSN was received (current time - LUSN timestamp.) When a node receives a location reply, it uses the age field to timestamp the LUSN with its own time minus the age.

If neither N nor any of N's neighbors have the location coordinates for destination D, N must revert to a selective flooding algorithm. N broadcasts a RREQ with the distance to the destination set to infinity. If an intermediate node receives a RREQ with the distance to the destination set to infinity, it will check to see if it has coordinates for the destination. If it does, it will forward the RREQ with its distance to the destination and the destination coordinates. If it does not, it will rebroadcast the RREQ with the distance to the destination set to infinity. This process is repeated until the destination is reached.

3.5.5 Route Error Messages (RERR)

Nodes mark routes as either active or inactive in the route table. A route may be deactivated for a number of different reasons. If a stored route remains unused after a certain timeout period, the route is de-activated. If a neighbor is removed from the neighbor table due to a broken link, all routes associated with that neighbor are de-activated. If data is received for a de-activated route, a route error message is constructed and propagated upstream toward the source, in the same fashion as a RREP. A RERR consists of the message type identifier and a route request sequence number. The RREQ_SN is included in the RERR message to identify the route that should be deactivated. When a node receives a RERR message, it deactivates the route associated with the specified RREQ_SN.

$$I2 \rightarrow I1: [[RERR,RREQ\_SN]I2\_k\text{-}]I1\_k$$

The RERR is signed with the sending node's private key and encrypted with the appropriate neighbor's public key. The appropriate neighbor is the neighbor listed in the reverse field of the route table for the specified route. When a node receives and successfully decrypts a RERR, it will update its routing table by marking the route associated with the RREQ_SN as inactive. If the node is not the source of the path

to be deactivated, it signs, encrypts, and transmits the RERR to the appropriate neighbor. When the source receives the RERR, it will deactivate the route and try an alternate if one is stored in its route table. If there isn't an alternate route or the alternate routes fail, the source re-initiates the route discovery process for the destination.

CHAPTER 4

DISCUSSION

4.1 Security

SPAAR provides the necessary elements to secure routing in a high-risk environment: authentication, privacy, and integrity. We now discuss how SPAAR satisfies the seven security requirements (see Table 3.1) that we feel are essential for a secure routing protocol in the targeted environment.

4.1.1 Security Requirements 1 through 5

The first five security requirements are satisfied through SPAAR's authentication techniques. A node may participate in SPAAR only after it has been authenticated by a secure certificate server and has received a public key certificate. In addition, each routing message in SPAAR is authenticated through the use of digital signatures. SPAAR consists of 7 routing messages: TBL_UPD, LOC_REQ, RREQ, LOC_REP, RREP, RERR and HELLO_REP.

The TBL_UPD, LOC_REQ, and RREQ are broadcast messages encrypted with a node's group encryption key (GEK). A group encryption key is a private key generated by each node dynamically. The corresponding group decryption key (GDK) is shared only with verified neighbors. The group decryption key is distributed to neighbors in a fashion guaranteeing authentication and privacy (see Section 3.4). The encryption of a message with a GEK is considered to be a digitally signed message. The successful decryption of a message with a GDK implies message authenticity.

The LOC_REP, RREP, and RERR are unicast messages signed with the private key of the sender. A node receiving such a message authenticates the message by decrypting it with the sending node's public key.

In SPAAR the HELLO message is not considered a routing message and is not authenticated. Each node periodically broadcasts a HELLO message as a way to allow new neighbors to be added to the neighbor table. HELLO messages contain only the sending node's public key certificate that has been previously signed by a certificate server, so there is no need to authenticate or encrypt it.

A node sends a HELLO_REP after receiving a HELLO message from an unknown neighbor. The HELLO_REP consists of the public key certificate, position coordinates, and transmission range of the node. The entire message is encrypted with the public key received in the HELLO message, and the position information and transmission range is also signed with the private key of the sending node. A node receiving a HELLO_REP must use the public key from the certificate to authenticate the signed portion of the message. As with the HELLO message, there is no need to authenticate the public key certificate since a certificate server has signed it previously.

4.1.2 Security Requirement 6

The exposure of network topology may assist an adversary trying to destroy or compromise nodes in the network, therefore network topology must be protected in a high-risk environment. SPAAR provides privacy between a node and its verified neighbors for each routing message. An unauthorized node will be unable to acquire topology information via routing messages on a SPAAR-protected MANET. We will now discuss how SPAAR provides privacy for each routing message to prevent disclosure of topology information to malicious nodes.

The group encryption key (GEK) used to provide authentication of TBL_UPD, LOC_REQ, and RREQ messages is also used to provide privacy between for these

messages between a node and its neighbors. The GEK is a private key and the corresponding GDK is public only to neighbors that have been authenticated and verified. When a message is encrypted with a GEK, only those neighbors possessing the GDK may decrypt the message. This prevents network topology, or any information that may be gathered from routing messages, from being exposed to malicious nodes.

The LOC_REP, RREP, RERR, and HELLO_REP are unicast messages that are encrypted with the public key of the receiving node. Properties of public key cryptography ensure that only the receiving node can decrypt routing messages, preventing network topology from being exposed to malicious nodes by routing messages.

4.1.3 Security Requirement 7

We have described the methods SPAAR uses to authenticate each routing message and provide privacy for routing messages between a node and its neighbors. These measures prevent most attacks by malicious nodes that result in nodes storing inaccurate routing information. Nevertheless, the replay of certain routing messages could trick a node into believing false (or outdated) routing information. In SPAAR, three routing messages in particular could be used by a malicious node to execute a replay attack: RERR, RREP, and TBL_UPD.

To prevent replay attacks, SPAAR utilizes sequence numbers. The location update sequence number, LUSN, is described in section 3.4. Each time a node sends a routing message containing position coordinates, such as a RREP or TBL_UPD, it includes a LUSN to represent the freshness of the coordinates. A larger LUSN indicates more recent coordinates. A node receiving a message with a LUSN will first compare it with the LUSN it has currently stored for that node. If the LUSN is larger than the one currently stored, the message is accepted and the new position

coordinates are stored. If the LUSN is less than or equal to the one currently stored, the message is assumed to be a replay and is therefore ignored.

The RREQ_SN is used to prevent replay of RERR messages. The RREQ_SN is incremented each time a node initiates route discovery and is included in each RREQ. When an entry is added to a node's route table as a result of a RREQ, the included RREQ_SN is stored. If a RERR message is received, the included RREQ_SN must match a RREQ_SN stored in the route table or else it is considered to be a replay and is ignored.

4.1.4 Non-colluding Compromised Nodes

Up to this point we have considered attacks originating from unauthorized malicious nodes. In a high-risk environment there may be a risk of attacks by compromised nodes that have been authorized and possess the appropriate cryptographic keys to participate in routing protocol.

In SPAAR, a compromised node may perform actions that can be considered violations of the security requirements. For example, a compromised node will be able to alter certain routing messages in transit, therefore SR2 will not hold. A compromised node will have access to topology information and location information of nodes, therefore SR6 will not hold. SR7 does not hold because a compromised node may lie about its position coordinates, causing another node to store inaccurate routing information. SPAAR has been designed to minimize the effects of malicious activity by compromised nodes and in most cases the effects are inconsequential. We present a set of attack scenarios to illustrate how SPAAR protects the network even in the presence of non-colluding compromised nodes.

Attack Scenario 1: In position-aided routing protocols a compromised node may attempt to distribute false position information about itself or other nodes. In SPAAR, a node may lie about its own position in an attempt to trick another node into believing it is a one-hop neighbor when it is actually out of range (perhaps

many hops away). A non-colluding compromised node could not successfully execute this attack in SPAAR, as HELLO and HELLO_REP messages are not forwarded or relayed by the receiver. A node out of range has no way to receive a HELLO or to transmit a HELLO_REP message without the aid of colluding compromised or malicious nodes.

Attack Scenario 2: A non-colluding compromised node may attempt to disrupt the network by generating false RREP messages, or alter RREP messages in transit to the source. In SPAAR, a RREP is only accepted by the source if it has been signed by the destination. The destination's public key certificate is included in the RREP, allowing the source (or any intermediate node) to authenticate the message.

Attack Scenario 3: In some MANET routing protocols [16, 13] the RREQ and RREP messages include a route record that includes the nodes the RREQ has traversed thus far. When a RREQ reaches its destination, the route record contains the entire path that the RREQ traveled. The route record is then included in the RREP and forwarded along the specified path by intermediate nodes. The route record divulges network topology information to an eavesdropper. SPAAR uses encryption to keep routing messages private to authorized nodes, however a non-colluding compromised node could eavesdrop and learn topology information from route records. To prevent this attack, SPAAR does not use route records. Instead, each node maintains forward and reverse entries in its route table as described in Section 3.5. In SPAAR, the routing messages do not divulge topology information, even to compromised nodes.

4.1.5 Colluding Compromised Nodes

It may be possible for multiple compromised nodes to collude (or conspire) to execute an attack. Two or more colluding compromised nodes can generally execute a wider range of attacks than attacks originating from non-colluding compromised nodes. These attack are more sophisticated and difficult to defend against.

Attack Scenario 4: Suppose a compromised node M1 receives a HELLO message from a legitimate neighbor N. The compromised node may forward the HELLO to another compromised node M2, not within range of N. M2 could then create a HELLO_REP containing false location information and send it to M1, who would forward it back to N. Node N would then believe M2 is a one-hop neighbor, when in fact M2 could be many hops away. This attack could be prevented if N has a method of verifying that the coordinates it receives from its neighbors are accurate, or if N has access to a trusted location service [18, 4].

4.2 Performance and Scalability

While SPAAR could use many position-aided routing protocols, currently SPAAR uses the widely accepted LAR protocol to make forwarding decisions in the route discovery process. We chose LAR due to its simplicity and its impressive performance evaluation. Simulation results of the LAR protocol indicate that using location information results in significantly lower routing overhead, as compared to a routing algorithm that makes no use of location information [5]. In addition, numerous optimizations are suggested that can further improve the performance of LAR. The reduction in routing overhead from the use of position information offsets the processing overhead caused by SPAAR's use of asymmetric cryptography.

An important factor in any MANET routing protocol is scalability. SPAAR has been designed to scale well. Some MANET table-driven routing protocols require that each node maintain up-to-date information about every node in the network. In SPAAR, a node is required to maintain up-to-date position information only for its one-hop neighbors and destinations for which it has active routes stored. This reduces the need for large amounts of non-volatile memory to store routing tables, even as the number of nodes in the network grows.

A node uses the group encryption key (GEK) to encrypt messages intended for its verified neighbors, such as RREQ and TBL_UPD messages. Since each of these

messages only needs to be encrypted one time, then broadcasted, as the number of nodes on the network increases, the computational overhead for the encryption of such messages remains constant, resulting in good scalability.

CHAPTER 5

CONCLUSIONS

5.1 Evaluation of Work Completed

In this paper we have presented SPAAR, a group of protocols that provides secure position-aided on-demand routing in a high-risk ad hoc environment. We have defined the seven security requirements that we feel are necessary for secure routing in a high-risk environment and shown how SPAAR satisfies these requirements. SPAAR protects position information with authentication, privacy, and integrity via public key cryptographic techniques. The protected position information is used to make forwarding decisions in the route discovery process resulting in a decrease in routing overhead. The position information is also used to verify one-hop neighbors. SPAAR only accepts routing messages from these one-hop neighbors, allowing us to prevent a number of well-known attacks on routing protocols.

SPAAR is capable of operating without an on-line PKI or key management system. The only requirements are that each node has a private key, is issued a public key certificate from a trusted certificate server, and has the public key of the trusted certificate server. The processing overhead of SPAAR is offset by the use of position information to make forwarding decisions resulting in a secure position aided ad hoc routing protocol for use in high-risk environments, with performance comparable to that of traditional MANET routing protocols.

5.2 Future Work

Future work may involve research in intrusion detection techniques to identify compromised nodes in MANETs. SPAAR defends against attacks from malicious unauthorized nodes, but is vulnerable to some attacks by compromised nodes. If an intrusion detection system could be designed to detect compromised nodes, SPAAR could exclude these nodes from the protocol. In addition, SPAAR could be modified to be more efficient, as it would no longer carry the burden of minimizing the potential for damage by compromised nodes.

Some position-aided routing protocols assume a decentralized location service [18, 5]. A location service provides the position coordinates of any node on the MANET. This assumption of a location service would significantly reduce overhead and increase the performance and scalability of SPAAR. The amount of non-volatile memory would be decreased because there would be no need to store the position coordinates of nodes.

SPAAR does not assume a location service for important reasons. For a location service to be appropriate for use in a high-risk environment, it must provide authentication, integrity, and privacy. Unfortunately, research in location services has not focused on security. A secure location service, to our knowledge, has not been designed. Future work may involve the design of a secure location service suited for use in a high-risk environment, to compliment SPAAR.

In SPAAR, a compromised node may attempt to lie about its location in an attempt to become a verified one-hop neighbor. We have shown that this attack is not possible unless two or more compromised nodes collude to execute the attack. Future work may involve investigating techniques that could allow a node to determine if another node is lying about its current position. For example, a voting scheme could be used among neighbors. Perhaps a measurement of propagation delay could assist in determining if a node is lying about its position.

SPAAR's primary concern is security. In order to achieve such a high level of security, SPAAR makes high use of asymmetric cryptography. We realize that asymmetric cryptography is resource intensive and future work may involve researching more efficient symmetric cryptographic techniques, to provide authentication and privacy in SPAAR. A performance analysis could then be done on SPAAR and the results could be compared with that of other secure routing protocols.

REFERENCES

[1] S. Corson and J. Macker. Rfc2501 mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations. Technical report, 1999.

[2] L. Zhou and Z.J. Haas. Securing ad hoc networks. In *IEEE Network Magazine, vol. 13, no.6*, pages 462–468, 1999.

[3] E. Royer and C-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. In *IEEE Personal Communications Magazine*, pages 46–55, 1999.

[4] J. Widmer M. Mauve and H. Hartenstein. A survey on position-based routing in mobile ad-hoc networks. In *IEEE Network Magazine*, 2001.

[5] Y. Ko and N. Vaidya. Location aided routing in mobile ad hoc networks. In *4th International Conference on Mobile Computing and Networking*, 1998.

[6] R. Morris and D. De Couto. Location proxies and intermediate node forwarding for practical geographic forwarding. Technical report, MIT Laboratory for Computer Science, 2001.

[7] J. Lundberg. Routing security in ad hoc networks. Technical report, Helsinki University of Technology, 2000.

[8] B. Karp and H. Kung. Greedy perimeter stateless routing for wireless networks. In *6th International Conference on Mobile Computing and Networking*, pages 243–254, 2000.

[9] V. Syrotiuk S. Basagni, I. Chlamtac and B. Woodward. A distance routing effect algorithm for mobility (dream). In *4th International Conference on Mobile Computing and Networking*, pages 76–84, 1998.

[10] John Marshall. An analysis of srp for mobile ad hoc networks. In *2002 International Multi-Conference in Computer Science*, pages 462–468, 2002.

[11] A. Perrig Y. Hu and D. Johnson. Wormhole detection in wireless ad hoc networks. Technical report, Rice University, 2001.

[12] E. Royer B. Dahill, B. Levine and C. Shields. A secure routing protocol for ad hoc networks. Technical report, University of Massachusetts, 2001.

[13] Papadimitratos and Z.J. Haas. Secure routing for mobile ad hoc networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2002.

[14] P. Naldurg S. Yi and R. Kravets. Security-aware ad-hoc routing for wireless networks. Technical report, UIU, 2001.

[15] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1999.

[16] D. Maltz D. Johnson and J. Broch. Dsr: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad Hoc Networking 2001*, pages 139–172, 2001.

[17] S. Callaghan and H. Fruehauf. Saasm and direct p(y) signal acquisition. In *GPS World Magazine*, 2002.

[18] D. Karger J. Li, D. De Couto and R. Morris. A scalable location service for geographic ad hoc routing. In *6th International Conference on Mobile Computing and Networking*, pages 120–130, 2000.

[19] Y. Zhang and W. Lee. Intrusion detection in wireless ad hoc networks. In *6th International Conference on Mobile Computing and Networking*, 2000.

[20] K. Lai S. Marti, T. Giuli and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *6th International Conference on Mobile Computing and Networking*, pages 255–265, 2000.

BIOGRAPHICAL SKETCH

Stephen H. Carter

Stephen Carter was born on July 10, 1978, in Jacksonville, Florida. He received his Associates in Arts degree from Florida Community College at Jacksonville in April of 1998. In December 2000, he received a Bachelor's degree (BS) in Computer Science from Florida State University. In December 2002, he will receive a Master's degree (MS) in Computer Science from Florida State University. He plans to pursue a career in the field of network security and eventually obtain a Doctoral degree in Computer Science or an MBA.