

# Secure Position Aided Ad hoc Routing

Stephen Carter  
Computer Science Department  
Florida State University  
Tallahassee, FL 32306-4530  
carter@cs.fsu.edu

Alec Yasinsac  
Computer Science Department  
Florida State University  
Tallahassee, FL 32306-4530  
yasinsac@cs.fsu.edu

## 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, as position information is broadcasted in the clear allowing anyone within range, including the enemy, to receive. We study methods of protecting position information in MANET routing protocols, and ways to use the position information to enhance performance and security of MANET routing protocols. We introduce "Secure Position Aided Ad hoc Routing" (SPAAR), a routing protocol designed to use *protected* position information to improve security, efficiency, and performance in MANET routing.

**Keywords:** Adhoc networks, Secure routing

## 1. Introduction

Mobile Ad hoc Networks (MANETs) are wireless networks with no 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 dependant 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*.

Research has shown that *position-based* routing protocols are a good alternative to on-demand protocols in many cases [4, 5]. Position-based routing protocols use node's geographical position to make routing decisions,

resulting in improved efficiency and performance. These protocols require that a node be 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 protect routing messages against malicious nodes and attacks that one could expect in hostile environments.

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 [6]. Position-based MANET routing protocols [4, 7, 8] 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 Secure Position-Aided Ad hoc Routing (SPAAR) as a method to protect position information in a high-risk environment. In Section 2 we discuss the target environment for SPAAR. Section 3 overviews related working secure routing and position aided routing. In Section 4 we present the details of SPAAR and follow up with a brief discussion and conclusion, in Sections 5 and 6.

## 2. SPAAR Environment

Due to the numerous applications of ad hoc networks, different ad hoc routing protocols must be designed and tailored for specific environments. SPAAR was designed for use in a specific environment. The targeted environment is a high-risk tactical MANET. A routing protocol is secure if it meets the security requirements for its environment.

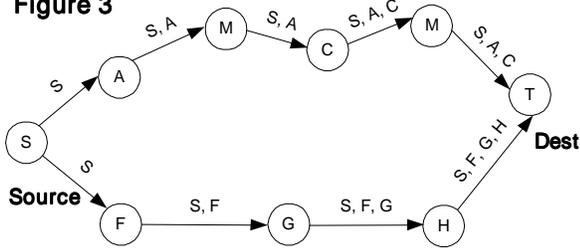
---

This material is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number DAAD19-02-1-0235.



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 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 3.

**Figure 3**



In Figure 3, the true shortest path (SFGHT) requires four hops. However the source node will not choose this path because it believes in the false path of 3 hops (SACT). 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.

This SRP attack clearly violates security requirement five that routes cannot be redirected from the shortest path by malicious action and security requirement seven, that the network topology must not be exposed neither to adversaries nor to unauthorized nodes by the routing messages.

In addition, SRP does not protect topology information. Route records are passed in the clear exposing the path a packet has travels thus far. Revealing topological information is unacceptable in a high-risk environment.

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. 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.

### 3.1.2 Security Aware Ad Hoc Routing

In [12], 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. This protocol uses 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 certain security attributes and trust

levels 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 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. SAR does not discuss how a node is assigned a trust value or how to 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 invisible node attack described in [11]. The framework provided by SAR is not sufficient to meet the security requirements of the high-risk environment that we target in this paper.

### 3.2 Position Based Routing Protocols

In topology-based protocols route discovery entails flooding a RREQ packet. In many cases, this technique is wasteful as the entire MANET is involved in a route discovery when only a small percentage of nodes, 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 improve performance over topology-based protocols in certain environments [13, 7, 8]. 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 decrease in routing overhead

### 4. Secure Position Aided Ad hoc Routing

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, we allow nodes to only accept routing messages from one-hop neighbors. The invisible node attack described in 3.1.1 and the wormhole attack [14], can be prevented if nodes only accept routing messages from one-hop neighbors.

In SPAAR, with the aid of position information, a node may verify its one-hop neighbors before including them in the routing protocol. SPAAR requires that each device can determine its own location. GPS receivers are relatively inexpensive and lightweight, so it is reasonable to assume that all devices in our network are equipped with one. In cases in which a node is unable to determine its location, nodes may use a *location proxy* as described in [5].

In SPAAR, the source node must also know the approximate geographic location of the destination. This may be calculated from the most recent location and most recent velocity information stored in the source node's destination table. If this is the source node's first attempt at

communication with a particular destination, the source may not have the destinations position. In this situation, a location service [15] may be used. If no location service is available, a selective flooding algorithm may be used to reach the destination and receive its position information.

## 4.1 Setup

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 nodes identity with it's 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.

## 4.2 The Neighbor Table

In 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. A node will only accept routing messages from a node in its neighbor table.

Specifically, each node maintains two keys for each neighbor. The first is the public key of the neighbor that is acquired from its certificate. The second is the neighbor's group decryption key that is used to decrypt RREQs, table update messages, and other routing messages encrypted with a group encryption key.

The position information is in the form of the neighbor's most recent location, represented as latitude, longitude coordinates, along with the neighbor's transmission range. Finally, each entry contains the neighbor's Table Update Sequence Number for use in the table update process.

### 4.2.1 Neighbor Table Creation

**Step 1:** A node N periodically broadcasts a "hello" message with its certificate. Nodes within range of N wishing to be recognized as neighbors decrypt N's certificate to verify and obtain N's public key. An entry for N is created in their neighbor table and N's public key is stored. Nodes respond with their certificate, coordinates, and transmission range encrypted under N's public key.

Upon receiving a hello response from a neighbor node X1, N verifies that X1 node is a one-hop neighbor. For all nodes that N verifies as one-hop neighbors, N stores the node's public key, most recent location, and transmission range in N's *Neighbor Table*.

**Step 2:** N generates a public/private key pair, which we call a *Neighbor Group Key pair*. The private part of N's neighbor group key pair is called *N's group encryption key* and denoted GEK\_N. The public part of node N's

neighbor group key pair is called *N's group decryption key*, denoted GDK\_N. N distributes its group decryption key to each of his neighbors listed in the neighbor table. The key is signed with N's private key to provide authentication, and encrypted under the neighbor's public key. Upon receiving the N's group decryption key, N's neighbors store it in their neighbor table.

It is important to note that at this point, X1 and X2 have the capability to accept routing packets from N, however they will not do so until they have *verified* N as a neighbor. This will occur after X1 and X2 broadcast a "hello" message and the above steps take place. This table state will last, at most, the time between "hello" broadcasts of X1 and X2.

### 4.2.2 Neighbor table maintenance

#### 4.2.2.1 Table update messages and TUSN

Each node periodically broadcasts a "table update" message to inform the neighbors of its new position coordinates and transmission range. Table update messages are encrypted with a nodes 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.

TUSN 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 TUSN prevents table update message replay attacks. In the RREQ a node uses the TUSN to inform its neighbors how fresh the coordinates are that it possesses for the destination.

When a table update message is received, the TUSN is time stamped allowing the node to determine how much time has passed since it has received a table update from its neighbors. After a timeout period has elapsed without a table update from a neighbor, the link is assumed to be broken and the neighbor is deleted from the table.

The interval at which a node broadcasts a table update depends on its mobility rate. A node with a high mobility rate broadcasts 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 & location request messages).

#### 4.2.2.2 Hello messages

All nodes broadcast periodic "hello" messages to add nodes to the neighbor table. 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 "NGK" field has a value. If the node has a value for node N's NGK 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 has no value for N's NGK field in the neighbor table, it sends a "hello response" message as

described above. As with table updates, the interval between hello messages is dependant on node mobility.

## 4.3 Route Discovery

### 4.3.1 Route Requests (RREQ)

**Step 1:** Node N broadcasts a RREQ with the RREQ sequence number, the destinations identifier, N's distance to D, D's coordinates and TUSN, all encrypted with its group encryption key. The RREQ sequence number is incremented each time a node initiates a RREQ. It is used to prevent replays of RREPs.

**Step 2:** RREQ recipients decrypt it with the appropriate group decryption key. Successful decryption implies that the sender of the RREQ is a one-hop neighbor. The identifier in the decrypted RREQ should match that of the neighbor whose group key was used to decrypt the RREQ.

**Step 3:** An intermediate node checks to see if it, or any of its neighbors, is closer to destination D. If an intermediate node has the destination's coordinates with a more recent TUSN, it uses those coordinates instead of the coordinates contained in the RREQ. If neither the intermediate node nor its neighbors are closer to the destination, the RREQ is dropped. If either is closer, the node forwards the RREQ with its identifier and distance to S, encrypted with its group encryption key. If the intermediate node had the destinations coordinates with a more recent TUSN, those coordinates replace the older coordinates in the RREQ. Intermediate nodes record in their route cache 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.

### 4.3.2 Route Replies (RREP)

**Step 1:** Upon receiving a RREQ, the destination constructs a RREP containing the RREQ sequence number, its coordinates, its velocity, and a TUSN. It then signs the RREP with its private key and encrypts it with the public key of the neighbor it received the RREQ from. The RREP propagates along the reverse path of the RREQ, being verified at each hop.

**Step 2:** 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, they setup forward entries in their route table that point to the node from which the RREP came. Intermediate nodes sign the RREP and encrypt it with the public key of the next node in the reverse route.

**Step 3:** The source node receives the RREP with the destinations location, velocity vector, and a TUSN. After successful decryption and signature verification, the source node verifies that the RREQ\_SN matches the RREQ\_SN from the initial RREQ. This prevents RREP replay attacks. If the RREQ\_SN is correct, the node updates its destination table with the new destination position information. As with table update messages, the source node time stamps the TUSN as an update history.

### 4.3.3 Route Error messages

Nodes keep track of *active* routes in the route table. A route may be deactivated for a number of different reasons. If a stored route remains unused for 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. The route error message is signed and encrypted along each hop, with intermediate nodes updating their routing tables to reflect the change. Upon receiving a route error message, the source may re-initiate the route discovery process for the destination.

### 4.3.4 Destination Table & Location Request

Each node maintains a destination table that contains a list of recent destination nodes it has communicated with. The destination table is identical to the neighbor table, except for the addition of the velocity field. Since a node will not receive update messages from nodes in the destination table, the *velocity* of destination nodes is recorded as a means of predicting the destinations current location. If the destination has an entry in the destination table, the time-stamped TUSN, MRL coordinates, and velocity can be used to compute an approximation of the destinations current location.

In the case that a node N does *not* have an entry for a destination in its destination table, a node broadcasts "location request" to its neighbors. Any neighbors that have the location coordinates for D will respond to S with a "location reply" encrypted with N's public key. Since SPAAR does not assume clock synchronization between nodes, the local timestamp on a TUSN is irrelevant to another node. Consequently, 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 TUSN was received (current time minus TUSN timestamp). When a node receives a location reply, it uses the age field to timestamp the TUSN with its own time minus the age.

## 5 Discussion

### 5.1 Security

SPAAR provides the necessary elements to secure routing in a high-risk environment: authentication, non-repudiation, confidentiality, and integrity. We now discuss how SPAAR satisfies the six security requirements for the managed-hostile environment [6].

**SR 1:** Fabricated routing messages cannot be injected into the network by malicious nodes. In SPAAR, Fabricated routing messages may include RREQs, RREPs, or table updates generated by malicious nodes. SPAAR provides authentication with each routing message to prevent fabricated messages from being injecting into the network.

**SR 2:** Routing messages cannot be altered in transit by malicious nodes. SPAAR calls for intermediate nodes to make changes to certain fields in the RREQ, such as the distance to the destination and the destination's coordinates. In SPAAR, only *authorized* nodes with the appropriate keys can make such changes.

**SR 3:** Routing loops cannot be formed through malicious action. Routing loop attacks may occur if a malicious node is able to spoof, or impersonate other nodes on the network. In SPAAR, each participating node is authenticated therefore impersonation is not feasible.

**SR 4:** Routes cannot be redirected from the shortest (or ideal) path by malicious nodes. Since only authenticated nodes can participate in SPAAR, and routing messages are only accepted from verified one-hop neighbors, malicious nodes are unable to redirect the ideal path.

**SR 5:** Unauthorized nodes should be excluded from route computation and discovery. SPAAR participants must possess a certificate signed by a trusted certificate server. Each node is authenticated and verified as a one-hop neighbor before it may participate in SPAAR.

**SR 6:** The network topology must not be exposed to adversaries or to unauthorized nodes by the routing messages. This security requirement is a matter of confidentiality and privacy in routing messages. For each routing message, SPAAR provides privacy between a node and its verified neighbors. A malicious node will not be able to acquire topology information via routing messages on a SPAAR-protected MANET.

## 5.2 Performance and Scalability

SPAAR uses asymmetric cryptography resulting in some processing overhead. However, SPAAR's overall routing overhead reduction, via geographic forwarding, offsets the processing overhead.

An important factor in any MANET routing protocol is scalability. SPAAR provides scalability comparable to that of other position aided ad hoc routing protocols. The primary difference between SPAAR and other position-aided protocols is SPAAR's use of cryptography. The cryptographic techniques employed by SPAAR should not affect scalability.

In SPAAR, nodes maintain up-to-date position information for its one-hop neighbors. A node uses a private group encryption key to encrypt messages intended for all of the node's neighbors, such as RREQs and table update messages. As the number of neighbors increase, the neighbor table grows. However the computational overhead for the encryption of messages remains constant.

## 6 Conclusion

In this paper we present a secure position aided ad hoc routing protocol. SPAAR is a routing protocol designed for a high-risk MANET environment. In particular, SPAAR satisfies the security requirements of the *managed hostile* environment. Our protocol protects position information with authentication, privacy, and integrity via

cryptographic techniques. The protected position information is used to reduce routing overhead and increase the security of routing, resulting in a protocol with performance comparable to that of traditional MANET routing protocols and secure enough for use in high-risk environments.

## Bibliography

- [1] S. Corson and J. Macker, Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, RFC2501, January 1999.
- [2] L. Zhou and Z.J. Haas, Securing Ad Hoc Networks, *IEEE Network Magazine*, vol. 13, no.6, December 1999.
- [3] E. Royer and C-K. Toh, A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks. *IEEE Personal Communications Magazine*, April 1999, 46-55.
- [4] Y. Ko, and N. Vaidya, Location Aided Routing in Mobile Ad Hoc Networks, *Proceedings of the 4th International Conference on Mobile Computing and Networking*, Dallas, USA, 1998.
- [5] R. Morris and D. De Couto, Location Proxies and Intermediate Node Forwarding for Practical Geographic Forwarding, Technical Report MIT-LCS-TR-824, MIT Laboratory for Computer Science, June 2001.
- [6] B. Dahill, B. Levine, E. Royer, and C. Shields, A Secure Routing Protocol for Ad Hoc Networks, University of Massachusetts Technical Report 01-37, 2001.
- [7] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward, A Distance Routing Effect Algorithm for Mobility (DREAM), *Proceedings of the 4th International Conference on Mobile Computing and Networking*, Dallas, USA, 1998, 76-84.
- [8] B. Karp and H. Kung, Greedy Perimeter Stateless Routing for Wireless Networks, *Proceedings of the 6th International Conference on Mobile Computing and Networking*, Boston, USA, 2000, 243-254.
- [9] Y. Zhang and W. Lee, Intrusion Detection in Wireless Ad hoc Networks, *Proceedings of the 6th International Conference on Mobile Computing and Networking*, Boston, USA, 2000.
- [10] Papadimitratos and Z.J. Haas, Secure Routing for Mobile Ad hoc Networks, *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, San Antonio, USA, 2002.
- [11] John Marshall, An Analysis of SRP for Mobile Ad Hoc Networks, *Proceedings of The 2002 International Multi-Conference in Computer Science*, Las Vegas, USA, 2002.
- [12] S. Yi, P. Naldurg and R. Kravets, Security-Aware Ad-Hoc Routing for Wireless Networks, UIUCDCS-R-2001-2241 Technical Report, 2001.
- [13] Y. Ko, and N. Vaidya, Location Aided Routing in Mobile Ad Hoc Networks, *Proceedings of the 4th International Conference on Mobile Computing and Networking*, Dallas, USA, 1998.
- [14] Y. Hu, A. Perrig, D. Johnson, "Wormhole Detection in Wireless Ad hoc Networks," Rice University Department of Computer Science, Technical Report TR01-384, 2001.
- [15] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris, A Scalable Location Service for Geographic Ad Hoc Routing, *Proceedings of the 6th International Conference on Mobile Computing and Networking*, Boston, USA, 2000, 120-130.