CERIAS Tech Report 2005-123

ON THE SECURITY OF DISTRIBUTED POSITION SERVICES

by Xiaoxin Wu and Cristina Nita-Rotaru

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

On the Security of Distributed Position Services

Xiaoxin Wu and Cristina Nita-Rotaru
Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

***Abstract*— Position-based routing protocols make routing decisions based on the geographical position of the destination of a packet. Such protocols scale well since they do not require nodes to maintain explicit routes. Instead each node must know only its own position, the position of its neighbors, and the position of the destination. Thus, a critical component of position-based routing protocols is the position service that allows nodes to obtain the position of a destination node.**

In this paper we analyze the security vulnerabilities of position-based routing protocols and virtual home region (VHR)-based distributed position service systems. We propose methods to protect the position information from both external and internal attackers. We then discuss and propose several mitigation mechanisms against position abuse by internal attackers that exploit the position service to trace their targets. Finally, we propose a position verification mechanism that allows the position service to verify that the positions reported by nodes are correct.

I. Introduction

A mobile wireless ad hoc network consists of a group of mobile nodes, which communicate with each other without any additional infrastructure support. Routing is challenging in such a network due to the lack of fixed infrastructure, as well as the node mobility and the dynamic network topology. Many routing protocols proposed for ad hoc wireless networks are on-demand in nature, such as AODV [1] and DSR [2]. In on-demand routing, a route to a destination node is established only when there is a need to route to that destination. The route discovery is initiated by the source which broadcasts a routing request in the entire network. This broadcast consumes significant bandwidth especially in large-size networks, and may cause a so-called ”broadcast storm” [3] problem in which the wireless channel is used mostly for control signaling.

One solution proposed to address the broadcast storm problem is using position-based routing protocols [4], [5]. Such protocols make routing decisions based on the geographical position of the destination of a packet. The approach does not require nodes to maintain explicit routes or to use broadcast for route discovery, resulting in increased scalability. Instead each node must know only its own position, the position of its neighbors, and the position of the destination. The position of the destination is carried along the route, such that a source or a forwarding node can determine its next hop locally, by selecting the closest node to the destination.

A node can obtain its own position through the Global Positioning Service (GPS) system [6], while its neighbors' positions can be obtained through a local information exchange. The position of the destination is usually obtained using a position service system that maintains the position information of all the nodes in the network. Any node can retrieve the position of another node using the position service. In an architecture where the ad hoc network is integrated with a fixed infrastructure such as a cellular-assisted ad hoc network [7], the position server can be attached to the fixed cellular network. The integrated architecture makes the position management, including the position update and position request/reply, less complex. However, in most cases, an ad hoc network is independent. Therefore, a position service system in which one or several ad hoc nodes act as position servers is more appropriate in such an environment.

Using one centralized position server for the entire ad hoc network is not practical because the server may be mobile and thus, it may not always be reachable by any node in the network. In addition, since a server is generally not more powerful than other normal node, it may become the operating bottleneck for the position management service. One way to address the above concerns is to use a distributed position service where several servers deployed in the network act as position servers. Every node has assigned a position server to which it must periodically report its position. Other nodes can retrieve the position of a destination node from the corresponding position server. A mechanism of a distributed position service system for mobile ad hoc networks based on a node's virtual home region (VHR), is presented in [8].

The transmission in the open medium, the autonomous nature of a node, and the routing dependence on unknown entities make an ad hoc network extremely vulnerable to attacks. Many attacks in ad hoc networks target the routing protocol [9], [10], by attacking the routing or the data packets. For example, an attacker can forge, modify, or replay routing packets, which can lead to discovering non-optimal or adversarial-controlled routes or can eavesdrop the data transmission, learning unauthorized information. An attacker can drop packets preventing routes from being established, or creating significant data loss in the network.

In addition the these attacks, positioning-based routing protocols are also vulnerable to new attacks targeting the position service. Since the position information is not protected, an attacker can use it to conduct more efficiently attacks such as eavesdropping, jamming, and wormhole. An attacker can also send false position reports to disrupt the position service and routing, or abuse the position service and misuse the position information to trace particular targets or learn the topology of the network. While secure routing in ad hoc networks received significant attention [9], [10], [11], [12], [13], less work studied the vulnerabilities of position service systems.

In this paper, we address security concerns in the context of position-based routing and position service systems. The major contributions are listed as follows:

- We identify the security vulnerabilities and possible attacks in positioning routing protocols and distributed position service systems.
- We propose a position verification mechanism that allows servers to verify the positions of reporting nodes and identify nodes who intentionally send false position information. The mechanism relies on polling and varies the transmitting power when sending the polling message to increase the accuracy of the position verification.
- We design mechanisms to protect the position information from either external or internal attackers who do not follow the position retrieving procedure. Our scheme has a low overhead by using both symmetric and public-key based cryptographic protocols to balance the communication and computation cost.
- We design a position misuse detection mechanism that constraints a node to use the position information obtained from the position service for routing only. The scheme identifies an internal attacker who abuses the position service system to trace potential targets.

The rest of the paper is organized as follows. In Section II we introduce the VHR-based distributed position service system. In Section III, we address security vulnerabilities regarding to position-based routing protocols. In Section IV we present the network and security assumptions used in this work. In Section V we give the details on the proposed security mitigating mechanisms for position verification, position protection and misuse. The related works are listed in Section VI, followed by the conclusion and future work in Section VII.

II. VHR-based Distributed Position Service System

In this section, we describe a distributed service system based on VHRs.

A. System Overview

In a VHR-based position service system, an ad hoc node is assumed to be able to obtain its own geographic position through position techniques such as GPS. Each node has a virtual home region (VHR) which is a geographical region around a fixed center. The relationship between a node identifier and its VHR center is given by a hash function. This function is predefined and known by all the nodes who join the network, so that other nodes can acquire a node's position by sending position requests to the corresponding VHR. The basic operations for a VHR-based distributed service system are illustrated in Fig. 1. In the figure, the requester stands for a node that needs the position of another node, the requestee stands for the node whose position is requested.

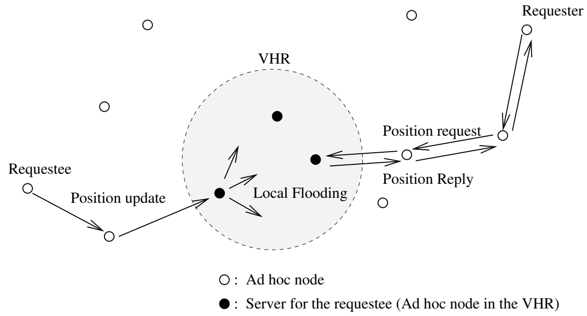

Fig. 1. Position management in a VHR-based position service system.

An ad hoc node updates its position when the distance between its current position and the latest reported position becomes greater than a threshold value. This threshold value is determined by the node density and the ad hoc radio transmission range [14]. The position update message is sent to all the servers residing in its VHR. Since the position of the VHR center is known, the routing between an ad hoc node and its position servers can be done also by using a position-based routing algorithm, by forwarding the position update message toward the VHR center. Once a server in the VHR receives the message, it will distribute the information to all the other servers via local flooding. The position update message is broadcasted once by every server in the VHR.

When a node, i.e., a requester, needs the position of another node, i.e., the position of the requestee, the requester will send a *position request* message to the requestee's VHR. The routing for the position request also is position-based. A routing request has to be received by only one position server who has the requested information. This server sends back a position reply following the reverse route.

The position servers are mobile nodes, thus the group of position servers for an ad hoc node is not fixed. A server moving into a node's VHR will become a new member of that group of servers, while a server moving out of the VHR will no longer serve that node. The VHRs for different nodes may overlap with each other. A node in the overlapped area will act as a server for multiple nodes.

B. AO2P: Routing Algorithm for Position Management

In a VHR-based position service system, position-based routing protocols can be used for position update and retrieve. However, position-based routing uses local position exchange among the neighboring nodes. This may lead to position information leakage. To address this problem, we use an ad hoc on-demand position-based private routing algorithm, named AO2P [14], to route the control messages of the position service system. AO2P forwards packets based on destination's position, which is carried in the route discovery message. As other geographic routing protocols, AO2P searches for routes in a greedy manner, i.e., a previous hop will find a node who can "move" its packet closest to the destination to be the next hop. Unlike other geographic protocols, in AO2P, a node does not need to report its updated positions to the neighbors, which is essential for position protection.

Since the previous hop does not know the positions of its neighbors, it can not decide its next hop directly. Instead, the next hop is determined by the contention among the neighboring nodes. In AO2P, neighbors of a previous hop are divided into classes of different priorities. A node closer to the destination has a higher priority, and normally wins the contention to become the next hop. Generally, the class for a node is assigned based on how close it is to the destination. There are special rules for node classifications. For example, upon receiving a position update, a node inside a VHR has a higher priority over a node outside a VHR so that the position update can be delivered to the VHR.

III. Attacks against Position-based Routing and Position Services

Traditional attacks in ad hoc routing, such as jamming, packet dropping, modification, fabrication or replay, and denial of service (DoS), can also be conducted against position-based routing algorithms. In addition, position-based routing protocols can be indirectly attacked, by disrupting the position service system or by taking advantage of the position service.

The position service also uses position-based routing for position management. Thus, attacks against the routing protocol targeting the position management messages can make the position service system to function incorrectly. As these messages are sent in plain text and without integrity verification, an attacker can change the position information carried in the position update or reply. This can allow an attacker to modify the identity of the requestee on a position request, so that a wrong position is sent back to the original requester. The destination position carried in a position update or request message has to be transmitted in plain text for routing purposes. An attacker can change the destination position carried in a position reply thus keeping the requester from getting the requested position. In general, these attacks can be addressed by using encryption, integrity and data authentication cryptographic mechanisms, as well as employing a routing protocol such as AO2P to route position management messages.

Not all attacks can be defeated by using cryptographic mechanisms. Examples include dropping packets and denial of service. An intermediate node may drop a position update sent by a node to its position server. This can result in stale position information that will prevent traffic to be routed to the new position of that node. An attacker can also continuously send out position requests to one or a few VHRs to keep the servers busy, so that other nodes can not access the servers.

In addition to attacks against the routing mechanisms themselves, position-based routing protocols are also vulnerable to attacks against the position management service. An attacker can use position information to stay close to a target, and conduct attacks such as traffic analysis, jamming, or wormhole attack [15], more efficiently. For example, an attacker that is not authorized to use the position service, referred to as an *external attacker*, can eavesdrop the channel and get a useful position by learning from the control packets which carry the position information. The position service can be exploited by a compromised node that is part of the network and authorized to obtain position information, referred to as an *internal attacker*. An internal attacker can abuse the position services by continuously sending position requests to the position servers to obtain the exact trajectory of its target or learn the network topology.

A node under adversarial control can also intentionally provide false positions. As position-based routing relies on correct positions, a false position of the destination will result in a routing failure. A false position from an attacker may make a neighbor to believe that the attacker is the closest to the destination. This neighbor then may select the attacker as the next hop and forward packets to it, allowing the attacker to obtain control of significant traffic.

IV. Assumptions

In this section, we present the network and security assumptions that are used in this work.

a) Network Assumptions: The network consists of a set of nodes and a set of servers that implement the position service. All nodes are uniformly distributed in the network, and move randomly with an average speed. Each node can obtain its own geographic position through GPS. Unless otherwise specified, all the nodes have the same transmitting power and consequently the same transmission range. The receiving range of a node is identical to its transmission range.

The position servers are deployed before any nodes join the ad hoc network. We assume that the wireless channel is symmetric and that the entire network is loosely synchronized, the clock drift being of the order of milliseconds.

b) Security Assumptions: An off-line certificate authority (CA) allows all nodes to obtain a pair of private and a public key (in the form of a certificate digitally signed by the CA). Each node has a pair of public and private keys used for authentication, non-repudiation, and symmetric key establishment.

The position servers are trusted and difficult to get compromised. The position servers also share a symmetric secret group key manually configured before the servers are deployed. The key is used to protect the confidentiality of the communication among the servers. This group key is periodically refreshed. All the servers also share a public/private key pair that identify the position service; the public key is used by any node to communicate with any of the position servers.

c) Attacker Model: An attacker is able to eavesdrop the communication channel, receive the packets within its receiving range, and drop, forge or modify packets passing through it. An attacker cannot compromise a position server.

We assume that an attacker does not have a stronger computing capability and a larger transmission range than a regular node. The attacker can jam the channel efficiently only if it is close enough to its target. It can not identify a node based on its transmission signatures, assuming that the identifier of the sender is not carried in the transmitted packet, or it is encrypted and can not be understood by the attacker. The attacker is not able to tell whether two transmissions are from the same sender.

Unlike the position servers, regular nodes can be compromised and under the control of an adversary, In this case, the adversary has access to all cryptographic keys stored by that node. Unless otherwise specified, at this stage, it is assumed that attackers do not collude and coordinate their attack.

V. Position Verification, Confidentiality, and Protection Against Misuse

This section presents mechanisms designed to protect the position service. We first describe in Section V-A methods that provide accurate position verification. Next, we discuss how to provide position confidentiality from an external attacker, or an internal attacker. Section V-B describes encryption and key establishment schemes for position update and retrieving. Finally, in Section V-C, we show how to prevent position service misuse by an internal attacker who can continuously request positions of its tracing targets, by using schemes that require proof of legitimate usage of requested positions.

A. Position Verification: A Polling Scheme

A basic approach for the servers to verify whether a position reported by a node is correct, is to send a message toward the reported position. Upon receiving a position update, the server replies to the sender with an acknowledgment. The acknowledgment will be routed to the sender via position-based routing using the reported position. A random number, referred as a *nounce*, is also included in the acknowledgment. The server accepts the position in the previous position update if the nounce is included in the following position update. Since the acknowledgment is sent immediately after the server receives the position update, it is unlikely that the tested node can not receive it due to a broken route between the server and itself. The only reason that the tested node cannot obtained the nounce is that it reported a false position, and based on this position, the acknowledgment cannot be delivered to it.

The server who first receives the updated message generates the nounce and distributes it with the updated position to other servers in the tested node's VHR, using the symmetric secret key shared by all servers. In this case, if another server receives the following position update from the tested node, this server can also verify the previously reported position. The verification result is then distributed within the VHR along with the updated position.

When a server sends the acknowledgment toward the updated position, it must include the destination position in the plain text, which is necessary for position-based routing. Therefore, the node's position is always disclosed to a number of nodes that are close to the route for the acknowledgment delivery. Sending an acknowledgment for every position update thus may lead to severe position information disclosure. To address the problem, we propose a *polling* position verification scheme. We also propose reducing the transmitting power for the polling message to improve the accuracy for the position verification.

1) **Random Polling Position Verification Scheme:** Instead of sending an acknowledgment upon every position update message, the server can send it after a random number of position updates. A testing nounce is included in the acknowledgment. We refer to this scheme as a polling scheme and to the acknowledgment as a polling message. The node who has been polled has to include the testing nounce in its next position update. The server may also require the polled node to reply right after the polling to get testing results quickly. However, this introduces more communication overhead.

It is possible that a malicious node sends a false position right after it has been polled. Since the probability that this node will be polled again is low, the false report may not be discovered. However, the false position can not be too far away from the real position, because the distance between this position and the position in the previous update, which is correct, should not be greater than a value[1].

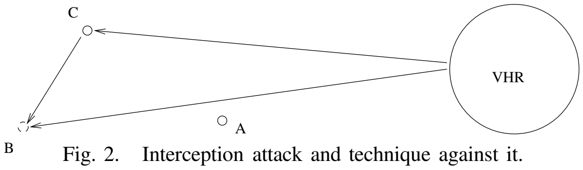

Fig. 2. Interception attack and technique against it.

Figure 2 presents a scenario where a malicious node reports a position that is on the extended line from its VHR to itself. As shown in Fig. 2, the malicious node at position A claims that it is at position B. When a server located in a VHR sends a polling message toward B, the malicious node can intercept the message. It then successfully sends a false position without being caught. We refer to this attack as the *interception attack*.

To defend against the interception attack, we propose to mask the polling message such that the attacker does not know that the position it reported is tested. For example, the tested position carried in the polling message may not be the exact position that is carried in its last position update, but a position close by. However, the attacker can still receive the message if it checks all the messages sent to the positions close to the false position it reported.

We propose a mechanism to mitigate the interception attack that does not pay the cost of the above method, by randomly selecting another node to perform the polling. The chosen node will receive the position that must be tested via a secure communication between the node and the position servers. As shown in Fig. 2, a polling message for testing a node at B is sent to a node at C first. The node at C then forwards the message toward B. The malicious node at A then cannot intercept the message. A server normally resides in the overlapped area of a number of VHRs and provides service to several nodes. It then is able to select the third party who is not close to the connection between the tested node and the VHR. The message is encrypted with the key shared between the third party and the servers (referring to Section V-B). In this case, even though the tested node can intercept the

[1] This value is the distance threshold value for position update in distance-based position update mechanisms.

polling message during the message delivery from the server to the third party, it cannot obtain the random number used for authentication and carried in the tested message.

It is possible that the route between the server and the third party, or the route between the third party and the tested node, does not exist. It is also possible that the third party is malicious and intentionally drops the test message. In both cases, the tested node is not able to receive the polling message making the server incorrectly conclude that the tested node provided a false position information, and diagnose it as a malicious node. To mediate this problem, when the server does not receive the testing nounce from the tested node, it polls a few more times using different intermediate nodes. The server makes a decision only when the tested node fails to reply to a number of polling messages.

2) **Position Verification Accuracy:** The proposed polling mechanism is able to catch a false position reporter if the position it reports is far away from its real position. However, a node can report a position with a relatively small error, such that when a polling message is sent to this reported position, the node can still receive it. As shown in [14], a packet delivered to a position can be received by a node even when located at half of the ad hoc radio transmission range away from that position. To catch a node that intentionally reports in-accurate positions, a polling message can be transmitted at a lower transmitting power. The lower the transmitting power is, the more precise a position verification can be.

However, lowering the transmitting power may lead to a higher number of hops of the path of the polling message, which subsequently results in a higher communication load. To address this problem, a server can send the polling message to the intermediate node using the normal transmission power. The intermediate node can be selected as the node closest to the tested position. This node then uses the reduced power to transmit the polling message. A power indication is carried in the message, according to which nodes who forward the message will use the same power.

Figure 3 shows the worst case scenario for the position verification accuracy when a malicious node reports a false position that is e away from its real position. We assume that the polling message is received by a node that is very close to the tested position. Since this node will broadcast the polling message, the malicious node can receive the polling message if the distance between its real position and the reported position is no more than r, the radio transmission range for the polling message. In addition, if the malicious node is positioned no more than r away from the path for the polling message, it can also receive the message, even if $e > r$. The area where a malicious position reporter can receive the message is the shaded area in Fig. 3.

The probability that in the worst case scenario, an attacker can be caught for reporting a false position because it cannot receive the position verification message, denoted as p_{cth}, is:

$$P_{cth} = \begin{cases} 0 & e \leq r, \\ 1 - \frac{arcsin(r/e)}{\pi} & e > r. \end{cases} \quad (1)$$

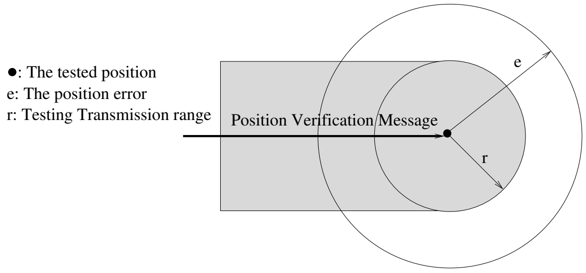

Fig. 3. Position verification for different transmission range values.

Figure 4 depicts the probabilities that a node who intentionally sends a false position can be caught in the worst case scenario. Different transmission power values for the polling message are used and therefore, different transmission ranges for the polling message, R_{test}, are obtained. The probability that a node reporting a false position can be caught increases as either R_{test} decreases or the position error, e, increases.

In Fig. 5, we show the simulation results for the probability of catching a false position reporter in the general case. According to both the analysis and simulation, it can be noticed that when the transmission range for the polling message is small, there is a great probability to catch a node who lies about its position even if this false position is very close to where this node actually is.

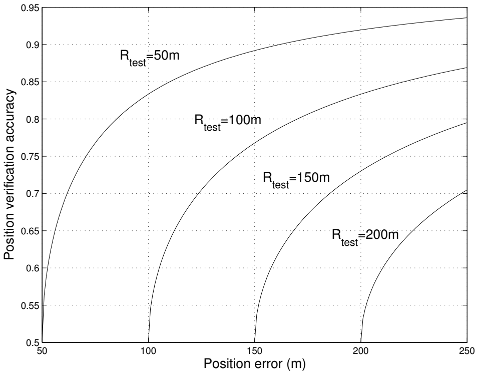

Fig. 4. Probability that a node sending false position can be caught in the worst case scenario.

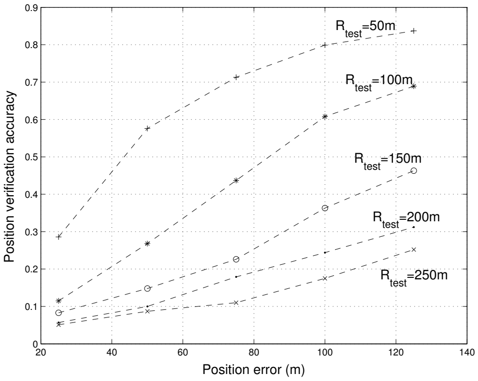

Fig. 5. Probability that a node sending false position can be caught.

B. *Position Confidentiality*

An attacker can learn the position of a node by eavesdropping the position management messages, such as the position update and the position reply messages. Encryption is needed to provide position confidentiality. In addition, a server has to make sure that a position update comes from a legitimate node with a known identity, while a requester has to make sure that the position reply is coming from a position server and not an attacker. This requires authentication and integrity of the messages exchanged by the nodes and the position servers.

A simple approach to achieve information confidentiality, integrity and authentication, is by using public key encryption, hash functions, and digital signatures. The control messages regarding to position management can be encrypted by either the shared public key of the servers or the public key of a node, depending on the direction of the communication. While less complex, this approach is too costly, particularly in the case of position update messages.

On the other hand, establishing a symmetric key between the servers and every ad hoc node for a position update and retrieve makes the cryptographic computation cost low. For every node, each server needs to obtain its key, once such a key is established by a negotiation between the node and any server, the key has to be flooded in the entire network so that every server can receive it and will be able to serve position requests. Such a flooding is extremely costly because it generates extensive communication overhead.

To keep both cryptographic and communication costs low, we propose a hybrid scheme, a symmetric key is used for distributing position update messages, while the public key is used for position retrieving. In the VHR-based position service system, a node and its servers have a relatively static relationship, i.e., a node always updates its position to a group of the servers in a fixed area. Therefore, the cost of establishing a symmetric key between a node and its servers can be amortized over many messages in case of position updates. On the other hand, since a requester only has a temporary relationship with the servers of its requestee, it is not efficient to establish the symmetric key for position retrieving. In this case, using public key encryption is more appropriate. As only one server needs to be contacted and reply to the requester, the corresponding cost of the public key process is not significant when compared with the cost when using public keys for position updates.

1) **Symmetric Key Approach for Position Update:** A node updates its position to the group of servers in its VHR. A symmetric key, defined as Traffic Encryption Key (TEK), can be used for position information confidentiality, integrity and authentication. As the position servers are trusted and can not be compromised, the group key management is less complex by excluding the needs that the TEK has to be revoked whenever a server leaves the node's VHR. The main procedures for TEK management are key establishment, key maintenance, and key revocation.

The definitions for some notations we will use in the rest of the subsection are listed in TABLE I.

TABLE I

SYMBOL DEFINITIONS.

$cert(i)$	i's certificate
E_k	encrypt with symmetric key k
E_{pk-i}	encrypt with i's public key
$sign_i$	sign with i's private key
h	hash function
$HMAC_k$	HMAC construction with key k

a) TEK Establishment: Since the servers are deployed before any node joins the ad hoc network, a newly joining node can contact the servers in its VHR to establish a TEK key. The joining node then sends its servers a message in the format:
$< position(VHR), distance,$
$E_{pk-s}(u, TEK, nounce, cert(u), position(u)), sign_u(h(msg)) >$.
$position(VHR)$ is the position of the center of this node's VHR. $distance$ is the distance from it to the VHR center. Once the position update is received by its next hop, the next hop will carry the distance from itself to the center of VHR in the position update and forward the message further. Both are carried in plain text for message delivery using the AO2P protocol (see Section II-B). The second part is the node's identifier u, the generated TEK, the nounce, and the position of the node. This part is encrypted by the server's public key. The last part is a signed hash of all the information carried in this message (denoted as msg), including the information transmitted in plain text. This part ensures message integrity and source authentication.

The server who receives the TEK initialization message will reply with an acknowledgment message. The message carries the nounce that was sent by the joining node and is sent back to u following the reverse route. The server will distribute the TEK and the updated position of u to the rest of position servers within the VHR, by encrypting and HMAC-ing [17] this information under the servers' symmetric shared key.

After the TEK has been established, the following position updates from this node will be in the format of:
$< position(VHR), distance,$
$E_{TEK}(position(u)), HMAC_{TEK}(msg) >$.
where msg denotes all the information in the message, including the information sent in clear.

b) TEK maintenance: Each server maintains the information for the nodes it serves, including nodes' identifiers, the positions for the served nodes' VHRs, the updated positions of the nodes, and the TEKs, in a TEK table shown in TABLE II. Since servers are mobile, a server may serve different nodes at different times. Consequently, its TEK table has to be updated. A server that just moves in a node's VHR has to obtain the TEK from the servers who have the key. Since the local flooding is used for position information distribution among all the servers in a node's VHR, the newly-coming server will

TABLE II

TEK TABLE.

Node ID	Position for VHR center	Updated position	TEK
A	(x_{VA}, y_{VA}, z_{VA})	(x_A, y_A, z_A)	TEK_A
B	(x_{VB}, y_{VB}, z_{VB})	(x_B, y_B, z_B)	TEK_B
C	(x_{VC}, y_{VC}, z_{VC})	(x_C, y_C, z_C)	TEK_C
...	...	...	...

receive the position update message encrypted with the TEK. When receiving a position update message a server will check its TEK table. If it finds out that it is in a new VHR and does not have the TEK for that position update, it will acquire the key from the other servers.

If at some point there is no server located in a node's VHR, then the TEK for that node will be lost and a newly-coming server cannot obtain the TEK to decrypt the position update message. In this case, the new server has to contact the node to obtain the old TEK or establish a new one. Since this newly-coming server cannot obtain the position information carried in the position update message, to contact the node it has to flood the TEK rebuilt message in the entire network. The message contains the position of the center of the VHR and is encrypted by the server's private key. After the node receives the message and confirms that the message is from a server, it starts a new TEK establishment process.

c) TEK revocation: There are two cases where a TEK revocation is needed. One case is when a TEK has been used for too long. In this case, either a server or the node can initiate a new TEK establishment using the public key infrastructure. Another case is when the servers have discovered malicious behavior of a served node and decide to stop providing position services for that node. In this case, the TEK revocation message will be distributed among the servers in the node's VHR using the symmetric secret key shared by the servers. Further position updates from the revoked node will no longer be accepted.

2) **Public Key Approach for Position Retrieving:** A requester sends a position request to a node's VHR only when it needs to know the node's position. After a requester contacts a server in the VHR, it may never re-visit this VHR again. The public keys are thus used for position retrieving. The message format for a position request is as follows:
$< position(VHR), distance,$
$E_{pk-s}(r, u, nounce, cert(r)), sign_r(h(msg)) >$.
where $position(VHR)$ is the position of the center of the VHR of the requested node u, $distance$ is the distance from the requester r or a forwarder of the message to the VHR center. The following part of the message are the requester identifier, the requestee identifier, the $nounce$, and the certificate. This part is encrypted by the server's public key. In the end of the message, the overall information is hashed and signed by r. This provides message integrity and message origin authentication. The server sends back a position reply, pos_{rep}, as follows:

$$< E_{pk-r}(u, nounce, position(u)), sign_s(h(msg)) > .$$

The routing information is not needed since the pos_{rep} can follow the reverse path of pos_{req}. The $nounce$ indicates that the position reply is linked to the right position request.

3) **Security and Overhead Analysis:** The position of a node in a position update is encrypted by the TEK shared by that node and the trusted position servers. Therefore, neither an external attacker or an internal attacker can obtain the position information by eavesdropping the position update message. Similarly, since a position reply is encrypted by a requester's public key, except for the requester, no other nodes can learn the requested position. TEK is also used to generate an HMAC of the content of the message proving the integrity and authenticity of the message. The use of digital signatures for the position request provides the requester's authenticity, which guarantees that only legitimate users can use the position services.

Cryptographic operations introduce an additional computation overhead on the position management protocol. Since symmetric keys are used for position update, the corresponding computation overhead is not significant. In contrast, the computation load for each position retrieving is high because of the use of public key encryption. However, each position retrieving implies a later on route discovery process. The position-based route discovery does not need to use flooding techniques. This significantly reduces the communication overhead, which compensates the cost of using public keys.

The communication overhead is mainly caused by the dissemination of position update messages. Each position update message has to be locally flooded. However, the area of a VHR normally is not large (especially when the server density is high). The corresponding communication overhead thus is not significant. A region-based local flooding mechanism in [16] can be used to further reduce the overhead introduced by position updates dissemination.

A large communication overhead will be generated when a TEK for a node is lost due to server mobility, and a server that newly enters the node's VHR has to regain the key. A network-level flooding has to be processed by that server to contact the node. Here we analyze how frequently this may happen in order to estimate the corresponding overhead.

We initialize the time when a previous position update is proceeded as 0. Assume that an old server leaves the node's VHR at t_d, a new server receives the following position update at t_u. This new server cannot obtain the TEK from the old server if $0 < t_d < t_u$.

Based on the results in [18], the time that a randomly moved unit may stay in an area can be approximated as exponentially distributed with a mean time of $\bar{t}$, and

$$\bar{t} = \frac{\pi S}{E[v]L}. \tag{2}$$

Here S and L are the area and perimeter respectively, and $E[v]$ is the average speed of the mobile unit.

A node updates its position to its VHR when the distance between its current position and the position reported in its last update is more than a threshold value. The time between any two consecutive position updates from a node is then equal to the time that this node stays in a circular area with a radius of d_τ. Define the time as t_u and its mean as $\bar{t_u}$, applying Eqn. (2),

$$\bar{t_u} = \frac{\pi d_\tau}{2E[v]}. \tag{3}$$

Similarly, the time that a position server stays in a VHR, defined as t_d, is also exponentially distributed with a mean of

$\bar{t_d}$. If the radius of a VHR is R_{VHR}, then:

$$\bar{t_d} = \frac{\pi R_{VHR}}{2E[v]}. \tag{4}$$

The probability that an old server is in the VHR and leaves before the next position update so that TEK needs to be re-established, denoted as p, is:

$$p = p[0 \leq t_d \leq t_u] = \int_0^{\infty} \int_0^{t_u} f_{t_u}(t_u) f_{t_d}(t_d) dt_d dt_u. \tag{5}$$

Due to the memoryless feature of exponential distribution,

$$f_{t_u}(t_u) = \frac{1}{\bar{t_u}} e^{-\frac{t_u}{\bar{t_u}}}, \tag{6}$$

and

$$f_{t_d}(t_d) = \frac{1}{\bar{t_d}} e^{-\frac{t_d}{\bar{t_d}}}. \tag{7}$$

Then:

$$p = \frac{\bar{t_u}}{\bar{t_u} + \bar{t_d}}. \tag{8}$$

If there are n servers that received the previous position update and have the old TEK, the TEK needs to be re-initialized only when all these servers leave the VHR before the arrival of the next position update. In such a case, the probability for a TEK re-initialization upon a position update is p^n.

Given a node density ρ, the probability that there are n nodes in an area S_0, defined as $P(n)$, is Poisson distributed, which is:

$$P(n) = \frac{1}{n!} (\rho S_0)^n e^{-\rho S_0}. \tag{9}$$

Let ρ_s be the density for servers, s_{VHR} be the area of a VHR. The probability for a TEK re-establishment at a position update, denoted as P_{tek}, is:

$$P_{tek} = \sum_{i=1}^{\infty} p^i P(i) = e^{-(1-p)\rho_s S_{VHR}}. \tag{10}$$

Let ρ_u be the density for ad hoc nodes. The TEK re-establishment frequency in a area unit, denoted as F_{tek}, then:

$$F_{tek} = \frac{\rho_u P_{tek}}{\bar{t_u}} = \frac{2E[v]\rho_u}{\pi d_\tau} e^{-(1-p)\rho_s S_{VHR}}. \tag{11}$$

We show in Fig. 6 the probability that there are no servers who have the TEK in the node's VHR and a network-level flooding is needed for a server newly entering the VHR to contact the node and re-establish the TEK. It can be observed that the server density and the size of VHR are the two major factors that impact this probability. When the server density is higher, or the size of VHR is larger, the probability of such a flooding is smaller. The reason is that more servers will be in the VHR, and the probability that the TEK can be maintained is larger. It is also shown in the figure that when the servers' density is high enough, the probability of a TEK re-establishment is very small and can be ignored.

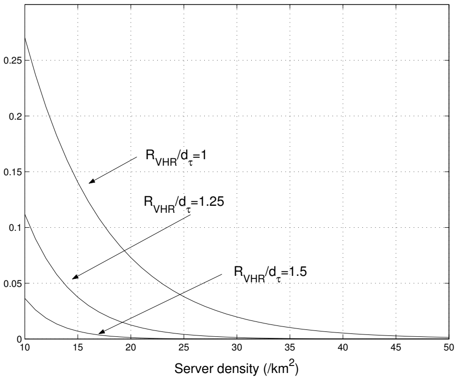

Fig. 6. Probability for TEK re-initialization upon a position update.

C. *Techniques Against Position Abuse*

A compromised node can take advantage of the position service and continuously send position requests asking for positions of other nodes for tracking purposes or to estimate network topology. We refer to such a use of position information as a position abuse.

We note that if the position information retrieved by a requester is used for legitimate communication, then the requester has to build a route from the requester to its requestee. Thus, the requestee can generate a *proof* for legitimate use of the position and can show it to the servers for further position retrievings. The proof can be a confirmation from the requestee showing that the requester has indeed built a route for meaningful communication with it within a reasonable time after the position retrieving. It is possible that a tracer can cheat by building up meaningless communications between itself and the requestee. It is the requestee's responsibility to judge whether the communication is meaningful.

Servers keep a record about when and by whom a node's position has been requested. This record is necessary for identifying the illegitimate position service users. The record is passed to the servers who newly enter the VHR, during the TEK distribution phase. Based on the record, a server that receives a position request determines whether a requester is a legitimate position service user and decides whether it should serve the requester. If not, the server will 1) deny the request; 2) put the requester on the questionable list; 3) alert the user; and 4) inform other servers in the entire network.

To prevent topology estimation, a user that does not show any proof can only request a small number of positions for different nodes. Since a position server may be in the VHRs of a number of nodes, the requests from an attacker for positions of different nodes may be received by the same server. It is less likely that a legitimate node sends a number of requests for the positions of different other nodes without showing any proof. Therefore, the abuser can be identified. A smart tracer can send requests to different VHRs that do not overlap. However, in this case, the topology information it obtains is not complete.

1) **Token-Based Abuser Identification:** The proof can be shown to the server by either the requester or the requestee. We describe a scheme, referred to as token-based abuser identification (TBAI), where the requestee shows the proof for legitimate position usage to the servers.

The position abuse is prevented by limiting the number of times a node can obtain the position of a certain node. Initially, a requester is assigned a number of *tokens* that allows it to request the position of a requestee. A server records how many tokens a requester has consumed for obtaining the positions of the nodes it serves. When a server receives a position request, it first checks the requester's token record. If the requester has not used up all the tokens, the server replies with the requested position and *activates* the counter for the token. The requestee includes the proof in its next position update message for token reimbursement, and sends the message to its VHR. Since the position update will be received by all the servers in the VHR, the server that was contacted by the requestee for the previous request will also receive the reimbursement message. It then *deactivates* the token counter. If the requester has used the tokens, the server will reset the number of consumed tokens to 0 and inform the other servers within the VHR. Otherwise there is no update for the tokens. If the position update message does not indicate a proof, the server will increase the token consumption by 1 and deactivate the counter. It then distributes this new record among all the servers in the requestee's VHR.

The token record needs to be distributed within a VHR only when the number of consumed tokens is increased or reset. In a network where most nodes are legitimate, these cases are rare. The storage consumption and communication overload of the scheme is low. In addition, no public key is needed for token-based scheme. The proof can be encrypted by using TEK, the symmetric key shared between the requestee and its position servers.

2) **Token Initialization:** Since the position reply is sent on the reverse route immediately after the position server receives the request, it is less likely that the route between the requester and the server is broken and consequently, the requester can not receive the position reply. However, it is possible that based on a retrieved position, a requester can not successfully build a route to the requestee. In this case, a proof cannot be provided. On the other side, the requester needs to send another request. A server will allow a requester to initially retrieve positions for a few times without requiring the proof.

The probability of a route discovery failure determines the lower bound for how many times a requester can retrieve a position without providing any proof. This number depends on the network topology and should be larger if the probability of a route discovery failure is higher, otherwise a requester may not be able to build a route to the destination at all. Let p_{fail} be the probability of a routing discovery failure, and n be the minimum number a requester can request a node position without showing any supporting evidence. The probability that any node can build a route to its destination before all the tokens are used up, defined as p_r, is: $p_r = 1 - p_{fail}^n$. Given the requirement for p_r, if p_{fail} is available, n can be calculated.

Figure 7 shows the probability that a node can not build up a connection with its destination before it tries n times. The simulation is conducted in a network covering an area of $1000m \times 1000m$, where the ad hoc nodes are uniformly distributed. The greedy geographic routing protocol is used for routing discovery. When n increases, the probability decreases. When $n = 3$, this probability is small and can be ignored. Simulation results also show that the probability of a routing discovery failure decreases as the node density increases. This means that the initial number of tokens depends on the node density, and can be a small value in a highly-dense network.

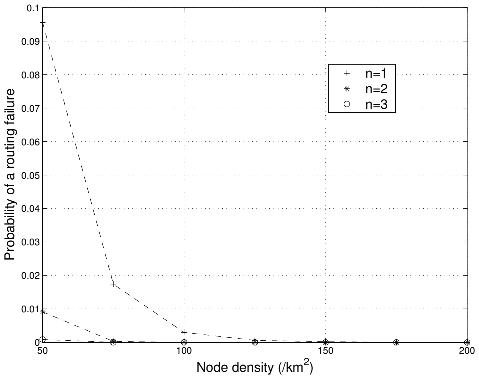

Fig. 7. Probability of a routing failure for different initial default tokens.

VI. Related Work

Below we review work in three areas related with the work presented in this paper: distributed position services, position verification, and key management in wireless ad hoc networks.

Distributed Position Services. In the Grid Location Service (GLS) [19], the area covered by the entire network is divided into an hierarchy of grids with squares of increasing size. In each level of the grids, a node is assigned an equal number of position servers. These servers have the closest identifier distance to this node's identifier, compared with all the other ad hoc nodes in the same grid. On the other hand, each node is a server for a number of other nodes, and has their updated positions.

In the Distributed Location Management (DLM) [20] service, the area covered by the network is also divided into an hierarchy of grids. Unlike in GLS, in DLM, for each node, its position servers are decided by whether the nodes reside in a certain area. The positions of those grids are the hashed result from the node's identifier, so that any other nodes who need this node's position know to which grid they should forward the position request to. A position server is selected by default, whenever it moves into a grid.

The proposed security mitigation mechanisms in this paper can be used in DLM with minor modifications. It is more difficult to apply the proposed security schemes in GLS because in GLS, whether a node can serve another node depends not only on the positions of the nodes, but also on the node identifiers.

*Position Security.*Several physical layer position verification schemes were proposed. In [21], verifiable multi-alteration

(VM) and verifiable time difference of arrival (VTDOA) are used to detect false positions, which enables secure computation and verification of the positions of wireless nodes in the presence of attackers. A number of reference points independently perform distance bounding to the verified wireless device. A centralized authority estimates the device's position based on the known positions of the verifiers and the distance bounds. VM prevents dishonest nodes from lying about their positions because of the property of distance bounding, that neither an attacker nor a prover can reduce the measured distance of the prover to the verifier, but only enlarge it. In [22], an echo scheme is used to check whether a wireless device is within a region. Both mechanisms can be applied jointly with higher layer verification algorithms to further improve verification accuracy.

Another attack on position systems targets the navigation signals or beaconing signals sent by the position reference points. The result is that the derived positions are not correct. Authentication schemes for such signals are proposed in [23] [24][25]. Another approach against the attack on beaconing signals is to use the redundant beaconing information. Examples include [26] and [27].

Key Management. A scheme that does not rely on a trusted third party (TTP) to public keys to nodes is studied in [28]. The scheme allows ad hoc users to generate public-private key pairs, to issue certificates, and to perform authentication regardless of network partitions and without relying on any centralized services. A performance study on both stateful and stateless group key rekeying algorithms is shown in [29]. The algorithms are analyzed in terms of storage cost and rekeying cost. In [30], the group key algorithms are applied to mobile networks and their performance is studied.

VII. Conclusion and Future Work

In this paper we focus on security vulnerabilities in virtual home region (VHR)- based distributed position service systems and propose security mitigation solutions. A polling scheme, where a server sends a message toward the position of the tested node, is used to verify positions. Sending the polling message through a randomly selected third node can defend against the interception attack. Simulation and analysis show that reducing transmitting power for the polling message can improve position verification accuracy. For position information protection, symmetric keys are established between nodes and their position servers. Analysis shows that while keeping the cryptographic computing load low, the key establishment and maintenance does not bring in significant control overhead. Finally, to prevent position abuse, a proof is required after a node has requested a position, which shows that this position has been used for building a route to the requested node. Our analysis of the initial number of the requests that can be processed without showing any proof shows that the number can be small if the network node density is high.

References

[1] C.E. Perkins and E.M. Royer, *Ad-hoc On-Demand Distance Vector Routing*, in Proc. of the 2^{nd} IEEE WMCSA, pp.90–100, 1999.

[2] D. Johnson and D. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, in Proc. of ACM SIGCOMM, 1996.

[3] Y. Tseng, S. Ni, Y. Chen, and J.Sheu, *The Broadcast Storm Problem in a Mobile Ad hoc Network*, in Proc. of MOBICOM'99, August, 1999.

[4] B. Carp and H. T. Kung, *GPSR: Greedy Perimeters Stateless Routing for Wireless Network*, in Proc. of MOBICOM'00, 2000.

[5] I. Stojmenovic, *Position based routing in ad hoc networks*, IEEE Communications Magazine, 40(7):128-134, 2002.

[6] B. Parkinson and S.Gilbert, *NAVSTAR: global Positioning System - ten years later*, in Proc. of IEEE, 1177-1186, 1983.

[7] B. Bhargava, X. Wu, Y. Lu, and W. Wang, *Integrating Heterogeneous Wireless Technologies: A Cellular-Assisted Mobile Ad hoc Networks*, Mobile Network and Applications, No. 9, 393-408, 2004.

[8] L. Blazevic, L. Buttyan, S. Giordano, J.-P. Hubaux, and J.-Y. Le Boudec, *Self-Organization in Mobile Ad hoc netwroks: The Approach of Terminodes*, IEEE Personal Communications, pp. 166–174, June, 2000.

[9] B. N. Levine, C. Shields, and E. Belding-Royer, *A Secure Routing Protocol for Ad Hoc Networks*, in Proc. of ICNP 2002.

[10] Y.-C. Hu, D. B. Johnson, and A. Perrig, *SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks*, in Proc. of the 4^{th} IEEE WMCSA 2002, June 2002, pp. 3-13.

[11] P. Papadimitratos and Z. Haas, *Secure routing for mobile ad hoc networks*, in Proc.of SCS CNDS 2002, January 2002.

[12] Y.-C. Hu, D. B. Johnson, and A. Perrig, *Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks*, in Proc. of MOBICOM'02, 2002.

[13] Y.-C. Hu, D. B. Johnson, and A. Perrig, *Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols*, in Proc. of ACM WiSe, 2003.

[14] X. Wu and B. Bhargava, *AO2P: Ad Hoc On-Demand Position-Based Private Routing*, in IEEE Transaction on Mobile Computing, 4:(4), 335-348, 2005.

[15] Y.-C. Hu, A. Perrig, and D. B. Johnson, *Packet Leashes: A Defence against Wormhole Attacks in Wireless Networks*, in Proc. of INFOCOM, 2003.

[16] X. Wu, *A Virtual-Home-Region Based Distributed Position Service System in Ad Hoc Networks*, In Proc. of ICDCS 2005.

[17] National Institute for Standards and Technology (NIST). *The Keyed-Hash Message Authentication Code*, FIPS 198, 2002.

[18] R. Thomas, H. Gilbert, and G. Mazziotto, *Influence of the Moving of the Mobile stations on the Performance of a Radio Cellular Network*, in Proc. of 3^{rd} Nordic Seminar, 1988.

[19] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris, *A Scalable Location Service for Geographic Ad Hoc Routing*, in Proc. of MOBICOM'00, 2000.

[20] Yuan Xue, Baochun Li and Klara, *A Scalable Location Management Scheme in Mobile Ad-hoc Networks*, in Proc. of the 26^{th} LCN, 2001.

[21] SrdjanCapkun and Jean-Pierre Hubaux, *Secure positioning of wireless devices with application to sensor networks*, in Proceedings of InfoComm, 2005.

[22] N. Sastry, U. Shankar, and D. Wagner. Secure, *Verification of Location Claims*, in Proc. of ACM WiSe, 2003.

[23] M. G. Kuhn, *An Asymmetric Security Mechanism for Navigation Signals*, in Proceedings of the Information Hiding Workshop, 2004.

[24] L. Lazos and R. Poovendran, *SeRLoc: Secure Range-Independent Localization for Wireless Sensor Networks*, in Proceedings of WiSe, 2004.

[25] Loukas Lazos, Radha Poovendran and Srdjan Capkun, *ROPE: ROBUST POSITION ESTIMATION IN WIRELESS SENSOR NETWORKS* in Proceedings of IPSN, 2005.

[26] Z. Li, W. Trappe, Y. Zhang, and B. Nath, *Robust Statistical Methods for Securing Wireless Localization in Sensor Networks*, in Proceedings of IPSN, 2005.

[27] D. Liu, P. Ning, and W. Du, *Attack-Resistant Location Estimation in Sensor Networks*, in Proceedings of IPSN, 2005.

[28] S. Caphun, L. Buttyan, and J.-P. Hubaux, *Self-Organized Public-Key Management for Mobile Ad Hoc Networks*, EPFL Tech. Rep., 2002.

[29] W. Chen and L. R. Dondeti, *Performance Comparison of Stateful and Stateless Group Rekeying Algorithms*, in Proc. of NGC, 2002.

[30] C. Zhang, B. DeCleene, J. Kurose, D. Towsley, *Comparison of Inter-Area Rekeying Algorithms for Secure Wireless Group Communications*, Performance Evaluation 49(1-4) 2002 1-20.