2005

# Po2V: Network Layer Position Verification in Multi-Hopo Wireless Networks

Xizoxin Wu

Cristina Nita-Rotaru
*Purdue University*, crisn@cs.purdue.edu

Report Number:
05-017

# PO2V: NETWORK LAYER POSITION VERICATION IN MULTI-HOP WIRELESS NETWORKS

**Xiaoxin Wu**
**Cristina Nita-rotaru**

**Department of Computer Sciences**
**Purdue University**
**Lafayette, IN 47907**

**CSD TR #05-017**
**July 2005**

# $Po^2V$: Network Layer Position Verification in Multi-Hop Wireless Networks

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

*Abstract*— **The correctness of the position is fundamental for position-based services. Previously proposed position verification schemes require an infrastructure, or the existence of a one-hop direct communication between the prover and the verifier. These schemes are less feasible in infrastructure-less networks where communication is achieved via multiple hops, such as mobile ad hoc networks. In this work we propose $Po^2V$, a lightweight, network layer position verification scheme for distributed position services in mobile ad hoc networks, under which a position server can verify whether a user has reported a correct position. We use adaptive transmitting power and multi-path polling to improve verification accuracy. We study different attacks against the verification scheme and propose corresponding mitigation techniques. We use a reputation system to reduce the false positives in a network with and without attackers. Our analysis and simulation studies show that while less expensive because no advanced techniques are required in the physical layer, $Po^2V$ can achieve a verification accuracy that suffices for many applications.**

## I. Introduction

Location-based services (LBS) [1], [2] have become an important part of pervasive computing services. A LBS server uses the location of a user to determine whether the user should be served. Location or position information is also used in mobile ad hoc networks to facilitate routing. Position-based (or geographic) routing protocols [3] make decisions based on the geographical position of the destination of a packet, and therefore have better scalability and routing performance (e.g., delivery ratio, end-to-end delay) than traditional routing protocols. In sensor networks, the location of a sensor determines whether the information collected by the sensor is useful, as the location indicates what area the sensor is monitoring.

The correctness of position is essential for position-related services. For example, location-based access control allows a client to obtain access to a service based on location. A malicious user may claim to be within an area that gives him fraudulent access to a service. In the context of sensor networks, many sensor applications require the knowledge of the origin of the sensed information. A malicious participant can pretend to be at critical positions and provide misleading information. In addition, a false position in sensor networks facilitates attacks such as sensor displacement (a sensor is temporarily moved out of the network by an attacker with the goal to compromise it, but the sink will believe that the sensor is still in the network), wormhole attack (a sensor thinks a node far away from it is its neighbor) and false network topology (a base station makes wrong routing decisions based on false sensor positions). In ad hoc networks, as position-based routing relies on correct positions, a false position of the destination will result in a routing failure. Moreover, an attacker can make a neighbor to believe that the attacker is the closest to the destination and be selected on the routing path, by manipulating the position information. As a result, the attacker will obtain control of significant traffic in the network. Finally, the correctness of the position information is essential for secure protocols that use position to prevent certain threats against network services [4], [5].

Position services can be classified in *network-centric* if the network obtains the position of the device directly, or *device-centric* if the method of obtaining the position relies on a device (i.e. a mobile LBS user, an ad hoc node, or a sensor node). For example, in a network-centric approach, the position of a user is learned based on the access point he is connected to or the base station he obtains service from, since he must be within their coverage area. More accurate network-centric position obtaining approaches can be found in [6], [7]. In comparison, in a device-centric service, the position of a device is obtained by the device itself, e.g., according to the navigation signals from GPS [8] satellites or the beaconing signals from pre-deployed landmarkers [9]. The position is reported to so-called position servers, which store the positions. Other network users can retrieve the position information from the position server for different network functions. Device-centric position service systems have been proposed in mobile ad hoc networks in [10], [11], [12], [13].

An attacker can generate false position information and interrupt network-centric position-based services by disrupting the position calculation function. In the case of device-centric position-based services, an attacker can simply report a false position or attack the navigation signal in GPS or the beaconing signal in sensor networks, ensuring that the positions obtained by honest users are not correct.

Position verification is a critical service. Its absence can result in numerous problems as discussed above. In the rest of the paper, we call the device of which the position needs to be verified a *prover*, and the parties that verify the position *verifiers*. Previously proposed position verification schemes [15], [14], [16], [17] require an infrastructure, because the

co-operation among a number of verifiers is needed to determine a false position report. Thus, they are less feasible in infrastructure-less networks, such as mobile ad hoc networks. In addition, as these schemes require one-hop connections between the prover and its verifiers, they cannot be applied directly in mobile ad hoc networks and sensor networks where the verifier can be several hops away from the prover.

In this paper we focus on attacks where a user intentionally sends false position reports in a multi-hop wireless network. We propose $Po^2V$, a lightweight, network layer polling based position verification scheme, that verifies the position information in a device-centric position-based service, in a multi-hop wireless network. Our scheme is less expensive because it does not use physical layer techniques [7] [18] [19] [20] to verify the position. The trade-off is that $Po^2V$ has higher granularity and it can verify that a prover is within an area [1]. The size of the area is determined by the radio transmitting range for the polling message. Such a verification result is sufficient for many scenarios. Examples of such scenarios are: region-based access services, where a server must verify that a mobile user is within a region; sensor monitoring, where what is needed is to estimate the area where the sensor is located, therefore to estimate the monitoring range; ad hoc geocasting [21], [22], where the region information of a destination is used to for routing.

We present and evaluate the scheme in the context of a distributed position service system designed for mobile ad hoc networks. We summarize the major contributions of the paper as follows:

- We design $Po^2V$, a network layer position verification scheme. A testing message, named polling message, is sent by the verifier toward the prover to verify the tested position.
- We evaluate the accuracy of the proposed scheme by both analysis and simulation. We propose as metrics the probability that a false reporter can be caught and the perturbation area that a false position can be located, to evaluate the accuracy. We also evaluate the false positive, where a legitimate position reporter is judged by mistake to be malicious.
- We propose and analyze different enhanced algorithms that further improve the scheme. We adapt the transmitting power of the polling message to reduce the testing error. We propose multi-path detection and advanced physical layer techniques to further improve the accuracy.
- We identify the potential attacks on the scheme and propose mitigating techniques. We address a so-called intersection attack and propose the triangle transmission through a third party to mitigate the attack. We analyze the attacks where a malicious party intentionally drops the polling message and propose a trust/reputation system to avoid blacklisting legitimate users.

To the best of our knowledge, the only work on multi-hop position verification is [23]. A verifier finds a number of position proxies that are within a one-hop connection with the prover, through which the prover's position is verified. $Po^2V$ is more general, the approach in [23] can be considered as a special case of our approach.

The rest of the paper is organized as follows. In Section II we present the related work. In Section III, we introduce the concept of using polling for position verification, followed by Section IV where we provide a detailed description of $Po^2V$. In Sections V, and VI, we analyze the accuracy for the position verification and show the major results of simulation study, respectively. In Section VII, we conclude the work.

## II. Related Work

### A. Position Service Systems

An example of network-centric position services is the $E911$ location system in a cellular network [6], where the cellular network determines the position of a cellular user by measuring the distance between the cellular user and a number of base stations. Another example is the indoor location system presented in [7]. A small unit called BAT is attached to the mobile object. The location system is fine-grained, with a number of fixed stations allocated in different areas in an indoor environment. Each time the fixed stations transmit a radio message containing a single identifier, causing the correspondent BAT to reply with a short pulse of ultrasound. The location system calculates the position of the BAT using the method of multilateration, according to the distance between the BAT and different fixed stations.

In a device-centric position service, the server can be a centralized entity, such as the base station in a sensor network or the cellular network in a cellular-assisted ad hoc network [31]. Servers can also be distributed, such as those in the typical works including Grid Location Service (GLS) [10], Distributed Location Management (DLM) (DLM) [11], and Virtual Home Region (VHR) based systems [12], [13].

In GLS, 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 assigns 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. Once a node needs the position information of another node, among all the nodes for whom it has the position information, it selects the node whose identifier is the closest to the target node and forwards the request. As in GLS, in DLM, the area covered by the network is 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 node who needs this node's position knows to which grid it should forward the position request to. The VHR-based position service system [13], [12], does not require the knowledge of grid hierarchy. Each node has a virtual home region (VHR) which is a geographical region around a fixed center. A number of servers are deployed in the network. A

[1] We note that physical layer techniques can be combined with $Po^2V$ in the last hop of the polling message path, to improve verification accuracy.

server located in a node's VHR will store the updated position of this node and respond to other nodes who request this position. 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.

### *B. Position Obtaining Techniques in Device-Centric Position Services*

A device can obtain its absolute position according to the distance to some reference points of which the absolute positions are known. The measure of the distance between a device and the reference points can be either range-based or range-free.

Range-based approaches use absolute point-to-point distance or angle information between the device and reference points to calculate the locations through multilateration. Common techniques for distance/angle estimation including Time of Arrival (TOA) [8], Time Difference of Arrival (TDOA) [18][7] [19], Angle of Arrival (AOA) [20], and Received Signal Strength (RSS) [18]. The well known GPS [8] system is a range-based system, where TOA is used for a GPS user to calculate its distance to a number of satellites, and therefore to obtain its own position. The range-based approaches result in accurate positions, yet the cost on hardware for radio, sound, or video signals is high. In addition, such approaches require strict time synchronization. These make the approach not suitable for the network built up by low-cost devices, such as sensor networks; or for dynamic networks, such as mobile ad hoc networks.

On the other hand, in range-free approaches, the position of an object is estimated by the connectivity between the object and the reference points. The exact distance or angle is not required. In [24], an approximate point in triangle (APIT) test algorithm is proposed. APIT resolves the localization problem by isolating the environment into triangular regions between anchor nodes. A node uses the point-in-triangle test to determine its relative location with triangles formed by anchors and thus narrows down the area in which it probably resides. APIT defines the center of gravity of the intersection of all triangles that a node resides in as the estimated node location. Other simple range-free algorithms can be found in [9] [25], under which location is calculated by finding the centroid of its proximate anchor nodes. The range-free approaches have less accurate results, yet the errors can be masked by fault-tolerance of the network, redundancy computation, and aggregation.

Range-free approaches have been extended so that an object can estimate its position even if it is multiple-hops away from reference nodes. Typical approaches [26], [27], [28] are proposed for sensor networks. In all these approaches, a position estimation is processed in three steps. In step 1, sensor nodes detect the local connectivity, i.e., find the other nodes that are within one hop from themselves, and then estimate the distance from themselves to the reference points, i.e., anchor node(s), based on the hop counts. In [26], the distance is obtained by accumulating the range for each hop. To address the problem that in a network the hop range depends on network topology, in [27] the range for a hop is obtained by finding out the average geographic distance for a hop in the multi-hop connections between anchor nodes. As the average hop range approach does not address the inaccuracy problem in highly irregular network topologies, in [28], a so-called Euclidian method is proposed, through which the local geometry of the nodes around an anchor can be deducted. Based on the information obtained from step 1, in step 2, the position of a node can be calculated through multilateration. However, as such a result may not be as accurate as that in the single-hop approach due to the error for the distance between the object and the anchor nodes, in multi-hop approaches, a refinement is needed in step 3. Such a procedure is iterative, until the positions are converged to a relatively stable value. A similar multi-hop position estimation approach that is designed for ad hoc networks is presented in [32], where the relative positions are estimated.

Unlike the multi-hop range-free approaches which rely on the hop count of the shortest path, making them highly vulnerable to different attacks on routing [29], [30], our scheme relies only on the polling message delivery and has no requirement for what path should be used. Therefore, diverse routes can be used to improve both the robustness against attacks and the verification accuracy.

### *C. Secure Position Systems*

Attacks can be conducted on beaconing signals sent from the reference points to either interfere with the signal or spoof the signal, so that an object can not derive its position correctly. Such an attack on the GPS navigation signal has been presented in [33]. Software changes in GPS receivers can mitigate some simple spoofing attacks [34]. For more sophisticated attacks, such as the so-called selective delay attack, an asymmetric security mechanism [33] can be used. Authentication for the beaconing signal in sensor networks can be found in [35]. In [35], a Secure Range-independent Localization (SeRLoc) scheme based on a two tier network architecture that achieves decentralized passive localization, is proposed. A symmetric key approach is used to authenticate the source of the beaconing signal. In addition, each reference point is equipped with directional antennas, covering different sector areas with different transmissions. A sensor detects the beacon signals from different reference points, computes the overlapping region of these beacon signals' coverage, and determines its location as the center of gravity of the overlapping region. The use of directional antenna improves the system robustness. A similar approach can be found in [36].

Another approach to defend against the attack on the beaconing signal is to use redundant beaconing information. A typical work is present in [37], where the redundancy at different levels in a wireless network is explored to tolerate, but not eliminate, the attacks. A robust statistical method is

proposed to make localization based on minimum mean square estimation (MMSE) less affected by different attacks.

The malicious beaconing detection is studied in [38] and [39]. In [38], a method is proposed to filter the malicious beacon signals on the basis of consistency among multiple signals. In addition, a scheme has been designed to tolerate the malicious beacon signal by adopting an iteratively refined voting scheme. In [39], a detecting node is used to discover the malicious beacon nodes. In particular, a round transmission time PDF function is used to filter out the malicious beacons.

Other than protecting beaconing signals, in [40], the knowledge of sensor deployment is used for a sensor to verify the position derived by itself. The main idea is that if a sensor is located at a position, it should have an expected neighborhood with sensors from different groups that have predictable geographic distributions.

In [15] Verifiable Multilateration (VM) is proposed, a technique that 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 [14], an echo mechanism is used to verify whether a wireless device is within a region. This is a rough verification that only determines whether a prover is within an area. [16], [17] show different ways of verifying the location of the satellite user. The network uses satellite ranges to estimate where its users are located.

## III. Position Verification by Polling

This section presents how the polling method can be applied to verify the position of a prover in a distributed position service system in mobile ad hoc networks. We also list the attacks that may undermine the polling-based verification scheme and list the cases of false positives.

### A. Network and Security Assumptions

We assume that ad hoc nodes are uniformly distributed in a specific area. A number of mobile servers are also distributed in the network. Each node has a virtual home region (VHR), which can be a circular area and the center of the area uniquely matches the node's identifier. Servers that are located within a node's VHR provide position services regarding to that node, such as position information storage, position update, and position retrieving. *Distance-based* position update is processed, under which an ad hoc node updates its positions to the servers in its VHR only when the distance between its current position and the position in its previous report is more than a threshold value. Other nodes obtain this node's position by contacting the servers. A server can be located in a number of VHRs and serve a number of nodes. Neighboring nodes exchange their position information. Therefore, a node knows the position of its one-hop neighbors. The Greedy Positioning Routing protocol (GPSR) [42], is used for position management message delivery.

We assume that there is an offline certificate authority that can assign public keys to the servers and ad hoc nodes. Servers are trusted and cannot be compromised. The communication between a server and its served ad hoc node as well as the communication between the servers is protected against eavesdropping, impersonation, modification, replay and injection attacks, as in [43]. In addition, a position reporter cannot repudiate its reports. We assume that malicious nodes are colluding and have advanced communication channels that allow them to share information.

### B. Basic Polling-based Position Verification 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 obtain 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 secure communication channel 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. This may not result in a direct position information exposure for the tested node. As the tested node can determine whether the testing message is destined for itself according to whether the position carried in the message is the position in its previous update, its identity does not need to be included in the polling message. However, sending an acknowledgment for every position update may lead to a match between the tested position and a node ID according to the moving trajectory of the tested node. In addition, sending a position acknowledgment upon every position update generates a significant overhead, unnecessary when most nodes are honest and report correct positions.

To address the problem, instead of sending an acknowledgment upon every position update message, the server can send it after a 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 polling scheme may work jointly with the existing position verification schemes that use physical layer techniques. A server may use these technique to verify the positions of the nodes that are within one hop of it, possibly by cooperating with other servers. If the nodes are more than one hop away, the server uses polling for position verification. Such a position verification scheme is illustrated in Fig. 1.

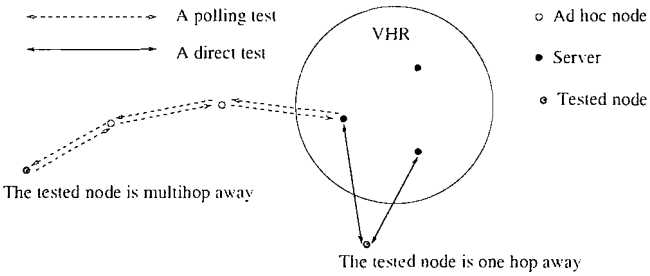



Fig. 1. A Polling-based Position Verification Scheme.

### C. Vulnerabilities and Attacks

When using the above basic polling scheme to verify a node's position, a malicious node is able to take advantage of the verification scheme and report a false position without being caught. We classify this type of attack as *deceiving position report attack*. On the other hand, malicious nodes can attack the polling messages. An honest position reporter therefore may not be able to receive the polling message and be mistakenly judged as malicious. We refer to this type of attack as a *blacklisting attack*.

*1)* **Deceiving Position Report:**

If a malicious node knows when the server will send a polling message (e.g., when server uses periodic polling), it can report the right positions at the time the server polls, and report false positions otherwise. Figure 2 shows a simple example. A server polls a tested node once every 3 position updates. A malicious node, who never moves, may claim its trajectory as shown in the figure. After the node reports its real position $L_1$, it can claim it is at positions $L_2$ and $L_3$ in its next two reports. These locations are $d_\tau$ away, where $d_\tau$ is the distance threshold value based on which a node has to update its positions. After that it reports its real position again, which is $L_1$. In this case, it can report a large number of false positions without being caught.

Another attack, which we refer to as *intersection attack*, takes place when malicious node can report a false position that is on the extended line from its VHR to itself yet still receive the polling message. As shown in Fig. 3, the malicious node at position $A$ claims that it is at position $B$. When a

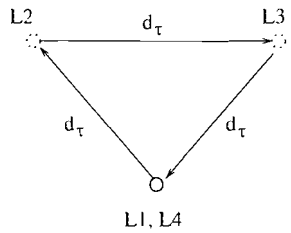



Fig. 2. Example of false position report.

server located in a VHR sends a polling message toward $B$, the malicious node can intercept the message and successfully send a false position without being caught. An extreme case of an intersection attack occurs when the malicious node stays within its VHR. If the size of VHR is not large, the malicious node can report a false position anywhere and intercept most of the verification messages sent from the servers. The intersection attack can severely interrupt the functionality of the position service system because a node can claim a position that is far away from its real position. If there are colluded nodes, a malicious node can mislead the position system by staying close to the servers of its partners and intercepting all the polling for them.

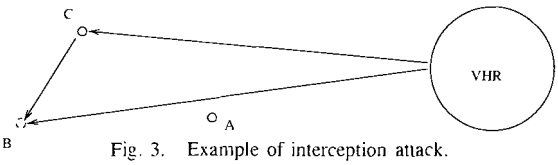



Fig. 3. Example of interception attack.

In both of the above attacks, a false position can be relatively far away from the real position. However, a malicious node can also take advantage of the in-accuracy of the polling verification method and claim false positions that are not very far away from its real position.

*2)* **Blacklisting Honest Nodes:**

In a polling-based verification scheme, the only trusted parties are the position servers and the routing for the polling messages relies on un-trusted intermediate forwarding nodes. Therefore, the scheme is prone to attacks against the multi-hop ad hoc connections between the verifiers and provers. Malicious nodes can interrupt the communication between a verifier and a prover so that once a polling message was sent out, the prover cannot receive it even if it has reported correct positions. As a result, the verifier will mistakenly judge an honest node to be malicious.

Traditional attacks in ad hoc routing, such as jamming, packet dropping, packet modification, fabrication or replay, and denial of service (DoS), can be conducted to attack the polling message. In addition, as position-based routing is used for polling message delivery, the scheme can be attacked by modifying the routing information, which is the position of the tested node. A node may also send out false positions during neighboring position exchanges. A malicious node can claim a position that is the closest to the destination so that a previous

hop may select it as the next hop. Once the polling message is sent to it based on the false position information, the malicious node can conduct further attacks such as packet dropping and manipulation. Attackers can also jam the area around a tested node or VHR, to cause DoS.

### D. False Positives

A false positive in a polling verification occurs when an honest user who has sent a correct updated position cannot reply to the polling message, not due to attacks, but due to failures of the poling mechanisms itself. False positives make the server incorrectly conclude that the tested node provided a false position information, and diagnose it as a malicious node.

For example, a false positive occurs if an honest user does not receive the polling message, because the route between the server and the tested node does not exist. This is possible especially when the polling message does not follow the reverse route of a position update. A tested node may not be able to receive the polling message also due to its own mobility. This may happen if the network is heavily loaded, and the delay for the polling message delivery is large. If the tested node is highly mobile, when the polling message arrives, the node may have already moved far away from its reported position and therefore cannot receive the polling message.

Another cause for false positives is due to the fact that radio propagation is hard to predict, and the estimated transmitting power based on positions may not be accurate enough. When using adaptive transmitting power for polling message transmission (as in IV-A.3.a), the transmitting power calculated based on the ideal channel propagation model may not generate the desired transmission range. A legitimate user who is located at its previously reported position thus may not receive the polling message.

Finally, although it has received the polling message successfully, a tested node fails sending the next position update because the route between itself and the server(s) at that time does not exist anymore.

## IV. $Po^2V$: Polling-based Position Verification Scheme

We propose $Po^2V$, a polling-based position verification scheme, under which a server (verifier) verifies the position reported by a tested node (prover) by sending a polling test message toward the reported position. A polling message carries the tested position in the plain text, and the nounce encrypted by the key shared between the server and the tested node. GPSR is used for polling message delivery. For the polling message delivery, each of the forwarders selects the neighbor that is the closest to the tested position as its next hop. Once the polling message reaches a node that is close enough to the tested position, i.e., the distance between this node and the tested point is no more than the ad hoc radio transmission range, this node becomes the *last hop* for the polling message delivery. The last hop broadcasts the polling message. If the prover has reported a correct position, it should be able to receive the message and return the nounce to the server.

In this section we provided a detailed description of $Po^2V$ that applies techniques to reduce false positives and mitigate the attacks previously described.

### A. Defending Against Deceiving Position Reports

*1)* **Random Polling:**

A tested node has to be polled randomly because a periodic polling gives a malicious node the opportunity to lie about its position without being caught. When the tested node is polled randomly, it is difficult for the malicious node to predict when its position will be verified. Therefore, there is a high probability that a false report is discovered.

*2)* **Triangle Verification:**

A solution to defend against the interception attack is 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 is not 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.

Another approach to defend against the interception attack is that when a tested node replies to the polling message, it includes the position of its previous hop from whom it receives the polling message. The previous hop signs its position and the neighborship between itself and the tested node. Thus, the tested node can not lie about its own position. This approach works against single attackers. When there are colluded malicious nodes, two attackers can claim false positions and neighborship to fool the server.

The attack from colluded attackers can be mitigated if the polling message is sent on a hop-by-hop path and the reply to the polling message includes the authenticated routing vector from the server to the tested node. As in the mechanism used for securing the BGP routing protocol [41], each node en route signs its existence and verifies the neighborhood relationship with the next hop. The router vector and the positions of the nodes en route are piggyback to the polling message. The tested node sends the information back to the servers. We note that this method, although will protect against colluding attackers, requires tremendous public key process for route vector authentication.

In this paper, 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 receives the position that must be tested via a secure communication between the node and the position servers. As shown in Fig. 3, 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 $A$ cannot intercept the message. A server normally resides in the overlapped area of a number of VHRs and serves several nodes. It 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 by the key shared between the third party and the servers. In this case, even though the tested node can intercept the 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. For example, a malicious node may intercept all the polling messages if it is located very close to the server. However, because it does not know the key shared between the server and the third party, it cannot decrypt the message and obtain the nounce.

*3)* **Improving 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 [44], a packet delivered to a position can be received by a node even half of the ad hoc radio transmission range away from that position.

We propose several techniques that improve the accuracy of the position verification scheme. As a result, a server will catch a false position reporter even though the position error is not large. The proposed techniques are using smaller transmitting power for the polling message, multi-path verification, crossing-layer verification, node trajectory, as well as geographic profiles.

*a) Adjusting Transmitting Power:* To catch a node that intentionally reports in-accurate position, a polling message is transmitted at a lower transmitting power. The lower the transmitting power is, the more precise the position verification is. The transmitting power is determined by the server, and a power indication is carried in the polling message, according to which nodes who forward the message will use the same power. In this paper, we refer to this scheme as fixed-transmission-power $Po^2V$ scheme, or $F\text{-}Po^2V$ scheme.

The trade-off of using a smaller transmission range is that lowering the transmitting power may lead to a higher number of hops during the polling message delivery, which subsequently results in a larger communication load. This load can be decreased if the polling message is sent to an intermediate node using the normal transmitting power, and in the rest of the route the lower power is used. However, if the distance between the intermediate node and the tested node is large, many extra hops will be involved.

To address the above problem, we propose another scheme, namely adaptive-transmission-range $Po^2V$ scheme, or $A\text{-}Po^2V$. The polling message is delivered using the normal transmitting power level. An intermediate forwarder that receives the polling message checks whether the tested position is within its transmission range according to the tested position and its own position, i.e., whether it is the last hop. If not, it forwards the message further. Otherwise it adjusts its transmitting power and makes its transmission range barely cover the tested position. This last hop of the tested node then sends the polling message using the adjusted power. As the distance between the tested position and the last hop is normally smaller than the maximum ad hoc transmitting range, the position verification result is more accurate than the scheme using maximum power.

*b) Multi-Path Parallel Polling:* To improve the position accuracy, more than one polling messages is sent to test a single position. The server selects the intermediate nodes at different positions, therefore the tested message is sent toward the tested position in different directions. Each testing message (polling message) carries a unique *nounce*. The position is verified only when the tested node shows that it has received all the nounces. The verification accuracy is improved at the price of communication overhead, because more polling messages have to be sent.

*c) Enhancing $Po^2V$ with Advanced Techniques:* A cross-layer design considering both network layer and physical layer mechanisms can provide powerful solutions for position security. For example, a position server may first authorize a mobile user that has a one-hop connection with the tested entity to process the verification. The mobile user uses the physical layer verification mechanism such as VM [15] to achieve better verification accuracy.

Applying smart antenna in the multi-path verification scheme is another approach to further improve the verification accuracy. A server selects two intermediated nodes such that two polling messages are sent toward the tested position in different directions, as shown in fig. 4. The verification accuracy depends on the width of the beam form. The approach will have an increased cost, because the cost on the physical layer increases.

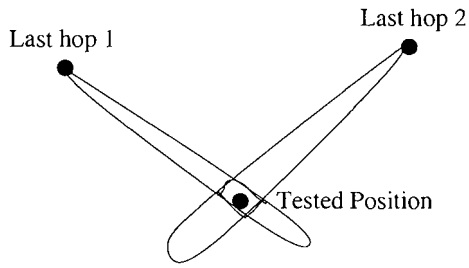



Fig. 4. The area where the tested node can receive both the testing messages.

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 can not be greater than a threshold value used in distance-based position update. The time between the two updates cannot be too short, otherwise the tested node has an unreasonable moving speed. A joint use of node trajectory and geographic profiles can help to detect the false positions. For example, if the node's previous position is on one site of a wide river, it is less likely it is on the other side of the river at its next position update if there is no bridge nearby.

### B. Mitigating Blacklisting Attacks

In this subsection we propose mechanisms that mitigate the blacklisting attack. In particular, we focus on the attacks where malicious nodes intentionally drop or manipulate the polling messages. Our solution does not focus on addressing attacks caused by jamming.

A simple solution to reduce the probability of blacklisting an honest user by interrupting the polling message delivery, is using multi-path delivery. A polling message is delivered through paths consisting of different intermediate forwarders. When the number of attackers in the network is small, it is less possible that there is an attacker in all of the paths so that all the polling messages are dropped. The server determines whether a node is honest based on statistics, i.e., the probabilities that a honest node responses to the polling when the network has different number of attackers.

A more sophisticated solution is building a reputation system for the users of the position service. The trust and reputation system in ad hoc networks has been studied in [45], [46], where the reputation of an ad hoc node is built upon both the first-hand observation and the second-hand observation. For example, for the reputation value of a node A at another node B, the first-hand observation is the behavior of $A$ that is detected directly by $B$, and the second-hand information is information obtained from any other nodes that have a reputation record for $A$. Such a system is complex because there is no trusted party. However, in our investigated scenario, all the position servers are trusted. This makes the reputation management less complex.

A server sets an initial value for the reputation of a user. The value will decrease if the user fails to reply a polling message. If a tested node replies to the polling message successfully, its reputation will increase. Once the reputation for a user drops below a threshold value, the server gives it a warning or prevent it from using the position services.

We denote $\gamma(n)$ as the accumulated reputation for a node whose position has been tested $n$ times. After the $(n+1)th$ verification, denote the updated reputation value as $\gamma(n+1)$, if the tested node replies the server with the *nounce* successfully,

$$\gamma(n+1) = \gamma(n) + \alpha. \quad (1)$$

Otherwise,

$$\gamma(n+1) = \gamma(n) - \beta. \quad (2)$$

The relationship between $\alpha$ and $\beta$ depends on the probability of a routing failure. For example, if the estimated probability for a routing failure in the network is $p$, to keep the reputation value of a legitimate user approximately as a constant, $\beta = \frac{1-p}{p}\alpha$.

The reputation value can also be used to decide how frequent a server should test a node. If a node has a low reputation, the server will test this node more frequently and vice versa.

A malicious node can take advantage of the reputation system with the help of colluding nodes. Suppose a malicious node always drops the polling message for the legitimate users, yet it forwards the polling message destined for a malicious node. In this case the probability of a routing failure is different between legitimate users and attackers. As an attacker has a lower probability of routing failure, the reputation system parameter designed for legitimate users may help a malicious user get a high reputation and send false positions while keeping its reputation value high. However, our simulation results show that such an attack can bring serious damage only when the number of attackers in the network is large. In addition, in $Po^2V$, it is difficult for a malicious forwarder to tell whether the tested node is an honest node or its colluding partner especially when the polling message is sent to the third party first, because only the tested position is carried in the plain text.

### C. Reducing False Positives

A false positive may occur if at the time the server sends a polling message toward the tested node, there is not route between them. 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 paths. The server makes a decision only when the tested node fails to reply to a number of polling messages. A complementary technique is to have the forwarder that detects a routing failure send back the information to the server. However, this generates more network control overhead. When multi-path $Po^2V$ is used, it is likely that the tested node does not receive all the messages. To reduce the possibility that it is blacklisted, the tested node sends back the partial testing information it receives.

Considering that a false positive is caused by node mobility, the problem can be solved by assigning the testing message and other position-related control messages (e.g., a position update message) with higher priority to access the wireless channel. When CSMA/CA is used for channel access, a higher channel access priority is achieved by using smaller-size contention windows or even shorter Differentiated Inter Frame Space (DIFS). Even if the network is highly loaded with data traffic, the testing message can still be delivered to the vicinity of the tested position in a short time after it updates the position.

To reduce the false positives caused by complicated radio propagation, a more sophisticated channel propagation model, such as the Walfisch-Ikegami model [47] developed for radio propagation in a metropolitan area, can be used. Fading and shadowing models should also be considered. In addition, geographic profiles such as the location of buildings may also be used.

In case the false positive is caused due to the routing failure for the polling reply, a mitigating mechanism is that once a tested node receives a polling message, it includes the nounce in a few of its following position updates. Another solution is that once a tested node receives a testing nounce, it updates its position immediately. As the time between the two transmissions is short, it is less unlikely that the route between the tested node and its VHR breaks.

## V. Analysis: Verification Accuracy

In this section we analyze the accuracy of the proposed position verification scheme using two metrics. One metric is the probability that a lying node can be caught when the false position it reports is a certain distance away from its real position. The other metric is known as the *perturbation area*. Perturbation area has been previously used to evaluate the location privacy in [48],[49]. In $Po^2V$, if a lying node stays in the perturbation area, it can still receive the testing message sent toward the false position. The larger this area is, the less accurate the testing algorithm can be. We use PDF function for the perturbation area to illustrate the verification accuracy.

In our analysis, we denote $(\widehat{r_1, r_2, d})$ as the curve when a circle with a radius of $r_2$ is cut by a circle with a radius of $r_1$, while the distance between the centers of the circles is $d$. We denote $S_{olp}(r_1, r_2. d)$ as the overlapped area of two circular areas with radiuses of $r_1$ and $r_2$.

### A. Single-Path Verification

In this subsection we analyze the verification accuracy by calculating the probability that a false position reporter can be caught.

*1)* **Verification with Fixed Transmitting Power:** Figure 5 (a) shows the case when fixed transmission range is used for polling message delivery, i.e., in $F\text{-}Po^2V$, and a malicious node reporting a false position is $e$ away from its real position. We analyze the case when nodes are uniformly distributed and node density is high. The delivery path for a polling message from a server or a third party to the tested position is approximately a straight line. Since the last hop for the polling message delivery 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 the radio transmission range for the polling message, which is denoted as $r$. 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. 5 (a). When the real position is $e$ away from the reported one, the false position reporter can receive the polling message only when it is located at the bold arch in the figure.

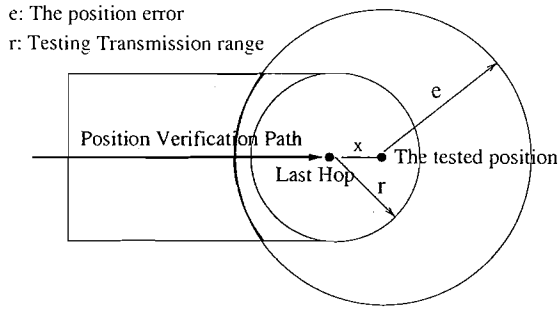
Let $x$ be the distance between the last hop and the tested position. The probability that an attacker can be caught for reporting a false position because it cannot receive the position verification message, denoted as $p(x, e)$, is:

$$p(x, e) = \begin{cases} 1 - \frac{(\widehat{r.e.x})}{2\pi e} & e \leq \sqrt{x^2 + r^2}. \\ 1 - \frac{arcsin(r/e)}{\pi} & otherwise. \end{cases} \quad (3)$$

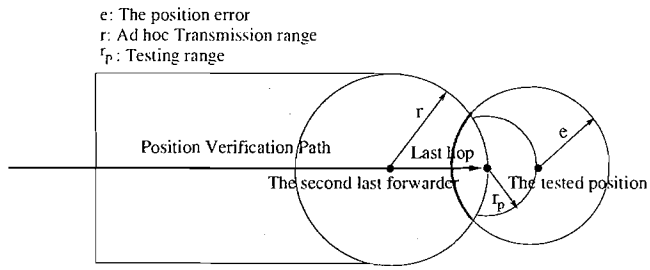
Given an error of $e$, the average probability that the malicious reporter can be caught, denoted as $P_{f-avg}$, is:

$$P_{f-avg} = \int_x p(x, e) f_x(x) dx. \quad (4)$$

$f_x(x)$ is the probability density function for $x$. In this case $x$ can be uniformly distributed between 0 and $r$. Although we cannot obtain an explicit function for Eqn. (4), $P_{f-avg}$ can be calculated numerically.




(a) Fixed transmission range




(b) Adaptive transmission range

Fig. 5. Position verification accuracy.

*2)* **Verification with Adaptive Transmitting Power:** Figure 5 (b) illustrates the case when adaptive transmission range is used to deliver the testing message, , i.e., in $A\text{-}Po^2V$, the scenarios that the malicious node can or cannot receive the testing message if its real position is $e$ away from the reported position. The bold arch represents the positions where the attacker can still receive the testing message. $r_p$ is the transmission range used by the last hop for the tested node. When adaptive transmission range is used, the probability that a lier can be caught is a function of $r_p$ and $e$, which can be denoted as $p(r_p, e)$. $p(r_p, e)$ can be formulated as:

$$p(r_p, e) = \begin{cases} 1 - \frac{(\widehat{r_p.e.r_p})}{2\pi e} & e \leq e_1, \\ 1 - \frac{(\widehat{r.e.r_p + r})}{2\pi e} & e \leq e_2, \\ 1 - \frac{arcsin(r/e)}{\pi} & otherwise. \end{cases} \quad (5)$$

$e_1$ and $e_2$ can be calculated in a straightforward way as follows:

$$e_1 = \sqrt{2r^2 - r_p^2 + \frac{r_p}{r}(r^2 - r_p^2)}, \quad (6)$$

$$e_2 = \sqrt{r^2 + (r + r_p)^2}. \quad (7)$$

Denote the probability density function for $r_p$ as $f_{r_p}(r_p)$. The average probability that a false position reporter can be

caught, denoted as $P_{a-avg}$, is:

$$P_{a-avg} = \int_{r_p} p(r_p, e) f_{r_p}(r_p) dr_p. \quad (8)$$

In this case $f_{r_p}(r_p)$ can also be assumed to be a uniformly distributed function between 0 and $r$. $P_{a-avg}$ then can be calculated numerically.

In the single-path verification scheme, the perturbation area is the shaded area in Fig. 5, which can be very large.

### B. Multi-Path Verification

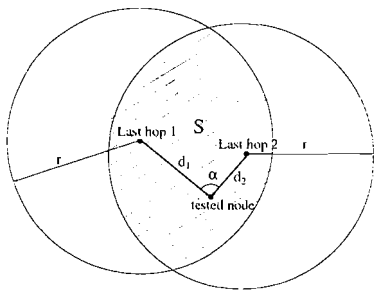
In this subsection we analyze the verification accuracy in the multi-path $Po^2V$. In particular, we calculate the probability that a false position reporter can be caught and the PDF of the perturbation area when tow polling messages are delivered to test the reported position.

*1)* **Verification with Fixed Transmitting Power:** The probability that a false position reporter can be caught when two polling messages are sent, denoted as $P_{f2}$, is:

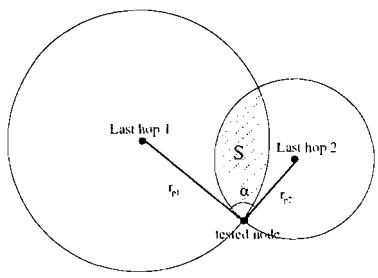
$$P_{f2} = 1 - (1 - P_{f-avg}^2). \quad (9)$$

where $p_{cth}$ can be obtained from Eqn. (3).

We now analyze the perturbation area. A typical scenario when two testing messages are delivered to the tested node in different paths is described in Fig. 6. The first testing message is finally received by *last hop* 1 and *last hop* 1 forwards the message toward the tested node. Similarly, the second testing message is finally received by *last hop* 2 and *last hop* 2 forwards the message toward the tested node. The perturbation area is the shaded area $S$.




(a) Fixed transmission range




(b) Adaptive transmission range

Fig. 6. The area that a tested node can receive both the testing messages.

We denote the distance between $F_1$ and $D$ as $d_1$, and the distance between $F_2$ and $D$ as $d_2$. We assume the server selects the third party randomly. Therefore, we can assume that $d_1$ and $d_2$ are randomly distributed between $(0, r)$, and $\alpha$ is randomly distributed between $(0, \pi)$.

The distance between $F_1$ and $F_2$, which is denoted as $d$, is formulated as:

$$d = \sqrt{d_1^2 + d_2^2 - 2d_1 d_2 cos\alpha}. \quad (10)$$

It is easy to derive the formula for calculating the area $S$, which is:

$$S = 2rarccos(\frac{d}{2}/r) - d\sqrt{r^2 - (\frac{d}{2})^2}. \quad (11)$$

Based on Eqn. (10) and Eqn. (11), and distribution functions of $d_1$, $d_2$, and $\alpha$, the distribution of $S$ can be calculated.

*2)* **Verification with Adaptive Transmitting Power:** The probability that a false position reporter can be caught when two polling messages are sent, denoted as $P_{a2}$, is:

$$P_{a2} = 1 - (1 - P_{a-avg}^2). \quad (12)$$

where $p_{avg}$ can be obtained from Eqn. (5).

When considering transmission range adaptation, the last hop 1 and 2 will adjust their transmitting power based on the positions of themselves and the position reported by the tested node. The transmission range is the distance between the last hops and the tested node. Similarly, the perturbation area $S$ can be calculated in forms of $d_1$, $d_2$, and $\alpha$, and therefore the correspondent PDF function can be obtained.

### C. Analysis Results

Figure 7 depicts the analysis results for the probabilities that a node who intentionally sends a false position can be caught when a single testing message is sent. We evaluate both the scheme using fixed transmission ranges (F-$Po^2V$) and the scheme with adaptive transmission range (A-$Po^2V$). In F-$Po^2V$, 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 A-$Po^2V$, the probability that a lier can be caught also increases as the error increases, but it increases slowly. The reason is that in A-$Po^2V$, a node sending a false report with an error can still have the opportunity to intercept the message if it is close to the message delivery path, and for the last forwarder (i.e., the last hop), other forwarders use the maximum transmission range, which is $250m$. However, the adaptive scheme can catch a lier with a high probability (more than 0.5) even if the error is very small. As shown in Fig. 8, if two messages are sent on different paths, the verification accuracy can be greatly improved.

Figure 9 shows the perturbation area in a multi-path $Po^2V$ schemes. For each verification, two polling messages are sent. The bold lines are the PDF values for $A$-$Po^2V$. It is observed

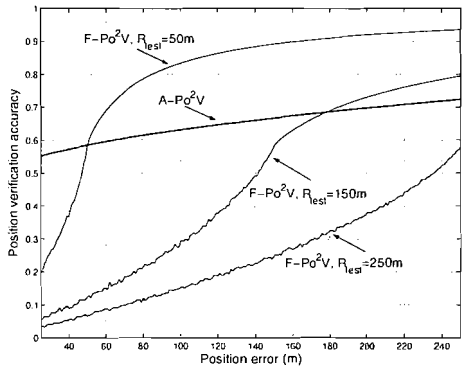



Fig. 7. Probability that a node sending false position can be caught when single polling message is sent.

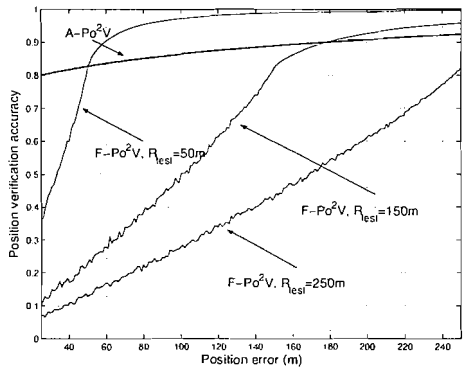



Fig. 8. Probability that a node sending false position can be caught when two polling messages are sent.

that for both perturbation area and perturbation strength, $A$-$Po^2V$ is comparable to the $F$-$Po^2V$ with the transmitting range for polling message set as $50m$, and better than the $F$-$Po^2V$ with larger transmitting ranges. Especially, there is a high probability that the perturbation area for $A$-$Po^2V$ is small. The perturbation area has a $50\%$ chance to be smaller than $2 \times 10^3(m^2)$.

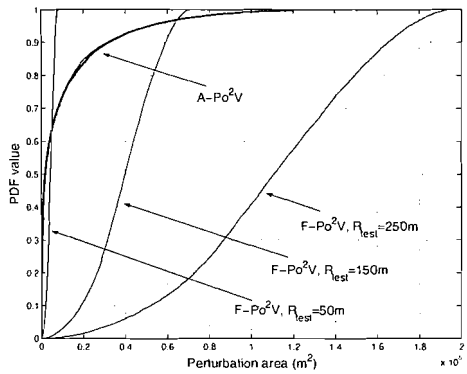



Fig. 9. The size of the perturbation area.

## VI. Simulation Study

To further evaluate the accuracy for the position verification scheme, we use simulation to collect the probabilities that a false position reporter can not be caught as well as the probability of false positives. The simulation scenario is a network with an area of $1500m \times 1500m$ where nodes are uniformly distributed. Unless otherwise specified, the transmission range for the polling message is $250m$.

In Fig. 10, we show the simulation results for the probability of catching a false position reporter in the general case. The results for the single-path $Po^2V$ are shown. It can be observed that when the transmission range for the polling message, $R_{test}$, 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. Similarly, Fig. 11 shows the improved capability of catching a false position reporter when multi-path $Po^2V$ is used. The verification results are close to those in the analysis.

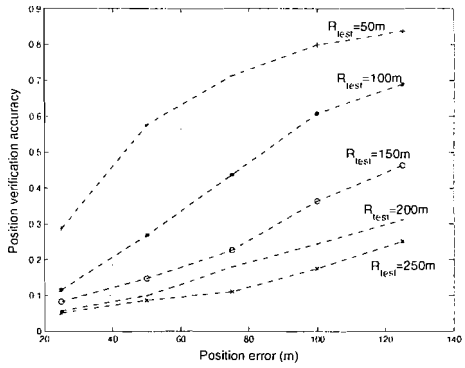



Fig. 10. Probability of discovering a false position reporter under single-path $F - Po^2V$.

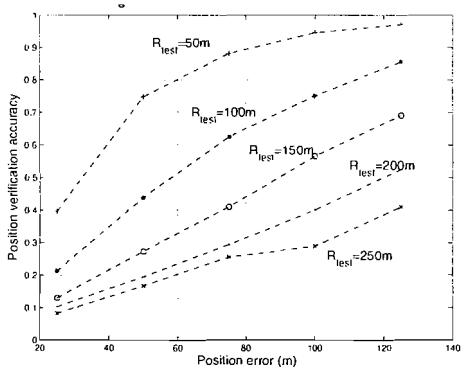



Fig. 11. Probability of discovering a false position reporter under multi-path $F - Po^2V$.

Figure 12 shows the simulation results when adaptive transmission range is used for polling messages. The results are similar to those in the analysis. A false reporter can be caught with a high probability even if its real position is close to the false position it claims. The multi-path scheme can further improve the verification accuracy.

Figure 13 shows the probability that when the server attempts to poll a tested node for $n$ times, it can not receive the polling message due to the failures of routing. The probability

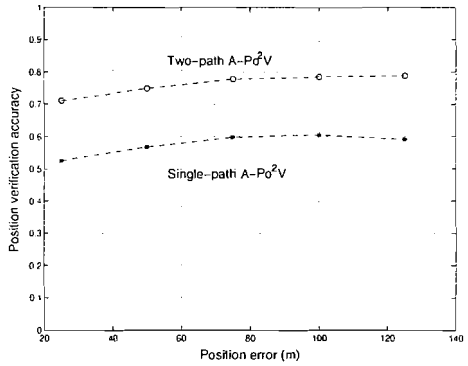



Fig. 12. Probability of discovering a false position reporter under $A-Po^2V$.

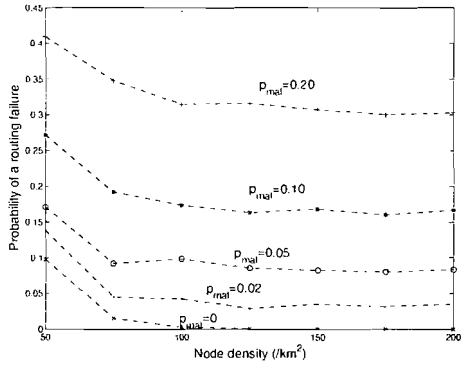



Fig. 14. Probability of a failure for the tested node to receive the testing message when there are malicious nodes in the network.

can also be viewed as the probability of a false positive if a server determines a node to be malicious when the tested node fails to reply to the polling message $n$ times. The greedy geographic routing protocol is used for polling message delivery. 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 in a highly-densed network, the server can decide that a node is sending false positions after it fails to reply to a small number of polling messages.

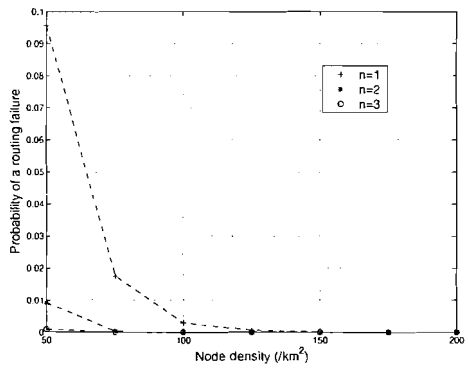



Fig. 13. Probability of a failure for the tested node to receive the testing message at different number of attempts.

Figure 14 shows the probability of a false positive, i.e., a tested node fails to receive the polling message when there are a percentage of $p_{mal}$ malicious nodes in the network. We assume that when a malicious node is assigned to forward the polling message, it always drops it. The curve on the bottom is the probability of a delivery failure when there are no malicious nodes in the network. We observe that when the number of malicious nodes increases, the probability of a failure increases. We also observe that when node density is high enough, the failure is mainly caused by intentional dropping. If there are a large number of malicious nodes in the network, the probability of a testing message delivery failure will be high and the position service system may not work.

Figure 15 shows the case when the proposed reputation system is used, the reputation value of a legitimate position service user. The network has a node density of $100/km^2$ and the percentage for malicious node is $5\%$. The initial reputation value is set as 1. According to the results in Fig. 14, the probability of a testing message delivery failure is 0.0985. Therefore, we select $\alpha$ value as 0.00197 and $\beta$ value as 0.01803 so that $\beta/\alpha = (1 - 0.0985)/0.0985$. The simulation results show that for relatively long period, the reputation for a legitimate user will maintain close to the original value.

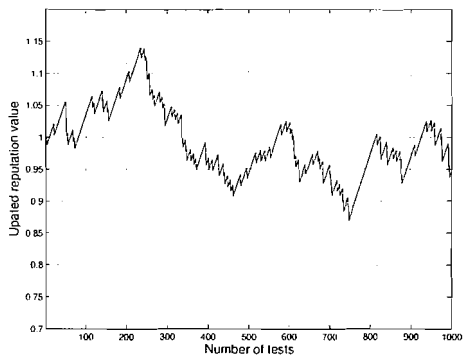



Fig. 15. Reputation for a legitimate user over a time period.

## VII. Conclusion and Future Works

We propose $Po^2V$, a lightweight, network layer position verification scheme in a distributed position service system designed for multi-hop mobile ad hoc networks. A server verifies whether a node has sent the correct position by sending a polling message toward the reported position using positioning routing. The verification accuracy improves when the transmitting power for the polling messages is reduced. To defend against interception attack, a triangle delivery is proposed. To mitigate the attacks on the polling message delivery and therefore the honest nodes can be blacklisted to be malicious, a reputation system is used so that a node is judged not based on its single behavior, but its behavior history. Different methods have been introduced to reduce the false positives caused by different reasons. $Po^2V$ is less expensive

than traditional verification schemes, easy to be implement, and provides a verification accuracy that suffices for many applications.

We will further study the robustness and accuracy for the verification scheme under colluding attacks, in a scenario where a large number of attackers are located around VHR. We will extend the scheme by adding the functionality that the servers within the one-hop of the tested node are used as the last hop for verification and combine it with a physical layer approach on the last hop to improve the verification accuracy.

## REFERENCES


[1] G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. *Cyberguide: A mobile context-aware tour guide*. ACM Wireless Networks, 3, 1997.

[2] *Computer Science and Telecommunications Board. IT Roadmap to a Geospatial Future*, The National Academics Press, November 2003.

[3] I. Stojmenovic. *Position based routing in ad hoc networks*. IEEE Communications Magazine, Vol. 40, No. 7, 2002.

[4] Y. Hu, A. Perrig, and D. Johnson, Packet Leashes. *A Defense Against Wormhole Attacks in Wireless Ad Hoc Networks*. in Proceedings of INFOCOM, 2003.

[5] D. Liu and P. Ning, *Location-based Pairwise Key Establishments for Static Sensor Networks*, in Proceedings of SASN, 2003.

[6] M. J. Meyer et al. *Wirelee Enhanced 911 Service-Making it a Reality*. Bell Lab Tech Journal, Autumn, 1996.

[7] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. *The anatomy of a context-aware application*, in Proceedings of MOBICOM, 1999.

[8] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*, Springer Verlag, $4^{th}$ ed., 1997.

[9] N. Bulusu, J. Heidemann, and D. Estrin. *GPS-Less Low Cost Outdoor Localization for Very Small Devices*, in IEEE Personal Communications Magazine, October 2000.

[10] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris. *A Scalable Location Service for Geographic Ad Hoc Routing*, in Proceedings of MOBICOM, 2000.

[11] Y. Xue, B. Li and K. Nahrstedt. *A Scalable Location Management Scheme in Mobile Ad-hoc Networks*, in Proceedings of IEEE Conference on Local Computer Networks (LCN'01), 2001.

[12] X. Wu, *VDPS: A VHR based Distributed Position Service System in Mobile Ad Hoc Networks*, in Proceedings of IEEE ICDCS, 2005.

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

[14] N. Sastry, U. Shankar, and D. Wagner, *Verifi cation of Location Claims*, in Proceedings of ACM WiSe, 2003.

[15] S. Capkun and J. Hubaux, *Secure Positioning of Wireless Devices with Application to Sensor Networks*, in Proceedings of INFOCOM, 2005.

[16] E. Gabber and A. Wool, *How to Prove Where you are: Tracking the Location of Customer Equipment*, in Proceedings of CCCS, 1998.

[17] E. Gabber and A. Wool, *On Location Restricted Services*, om IEEE Networks, Nov./Dec., 1999.

[18] P. Bahl and V. N. Padmanabhan, *RADAR: An In-Building RF-based User Location and Tracking System*, in Proceedings of INFOCOM, 2000.

[19] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, *The Cricket Location-Support System*, in Proceedings of MOBICOM, 2000.

[20] D. Niculescu and B. Nath, *Ad hoc Positioning System (APS) Using AoA*, in Proceedings of INFOCOM, 2003.

[21] K. Chen and K. Nahrstedt, *Effective location-guided tree construction algorithms for small group multicast in MANET*, in Proceedings of INFOCOM 2002.

[22] Young-Bae Ko and Nitin H. Vaidya, *Geocasting in Mobile Ad Hoc Networks: Location-Based Multicast Algorithms*, in Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications, 1999.

[23] B. Waters and E. Felten, *Secure, Private Proves of Locations*, Princeton University, Tech. Rep.

[24] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. F. *Abdelzaher, Range-free Localization Schemes in Large Scale Sensor Networks*, in Proceedings of MOBICOM, 2003.

[25] N. Bulusu, J. Heidemann, and D. Estrin, *Density Adaptive Algorithms for Beacon Placement*, in Proceedings IEEE ICDCS, 2001.

[26] A. Savvides, H. Park, and M. Srivastava, *The Bits and Flops of the N-hop Multilateration Primitive for Node Localization Problems*, in Proceedings of ACM WSNA, 2002.

[27] D. Niculescu and B. Nath, *DV Based Positioning in Ad Hoc Networks*, in Telecommunication Systems, Vol. 22, No.1, 2003.

[28] C. Savarese, K. Langendoen, and J. Rabaey, *Robust Positioning Algorithms for Distributed Ad-hoc Wireless Sensor Networks*, in Proceedings of USENIX Technical Annual Conference, 2002.

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

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

[31] 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, 2004.

[32] S. Capkun, M. Hamdi, and J.-P. Hubaux, *GPS-free Positioning in Mobile Ad-hoc Networks*, Cluster Comput, Vol. 5, No. 2, 2002.

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

[34] J. S. Warner and R. G. Johnston, *Think GPS Cargo Tracking High Security? Think Again*, Technical report, Los Alamos National Laboratory, 2003.

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

[36] L. Lazos, R. Poovendran and S. Capkun, *ROPE: Robust Position Estimation in Wireless Sensor Networks* in Proceedings of IPSN, 2005.

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

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

[39] D. Liu, P. Ning, and W. Du, *Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks*, in Proceedings of IEEE ICDCS, 2005.

[40] W. Du, L. Fang and P. Ning, *LAD: Localization Anomaly Detection for Wireless Sensor Networks*, in Proceedings of IPDPS, 2005.

[41] S. Kent, C. Lynn, J. Mikkelson, and K. Seo, *Secure Border Gateway Protocol (S-BGP)- Real World Performance and Deployment Issues*, in Proceedings of NDSS, 2000.

[42] B. Carp and H. T. Kung, *GPSR: Greedy Perimeters Stateless Routing for Wireless Network*, in Proceedings of MOBICOM, 2000.

[43] X. Wu and C. Nita-Rotaru, *On the Security of Distributed Position Services*, in Proceedings of SecureComm, 2005, to appear.

[44] X. Wu and B. Bhargava, *AO2P: Ad Hoc On-Demand Position-Based Private Routing*, in IEEE Transaction on Mobile Computing, Vol. 4, No. 4, 2005.

[45] S. Buchegger and J. Y. Le Boudec, *The Effect of Rumor Spreading in Reputation Systems for Mobile Ad-hoc Networks*, in Proceedings of WiOpt, 2003.

[46] S. Buchegger and J. Y. Le Boudec, *A Robust Reputation System for P2P and Mobile Ad-hoc Networks*, in. Proceedings of the $2^{nd}$Workshop on the Economics of Peer-to-Peer Systems, 2004.

[47] E. Damosso and G. De Brito *Mobile Radio: COST 231 View on the Evolution towards 3rd Generation Systems*, in Final Report of the COST 231 Project, 1998.

[48] B. Gedic and L. Liu, *Location Privacy in Mobile System: A Personalized Anonymization Model* in Proceedings of IEEE ICDCS, 2005.

[49] R. Cheng, D. V. Kalashnikov and S. Prabhakar, *Querying Imprecise Data in Moving Object Environments*, in IEEE Transactions on Knowledge and Data Engineering (IEEE TKDE), Vol. 16, No. 9, 2004.