RESEARCH ARTICLE OPEN ACCESS

Identification of Closest and Phantom Nodes in Mobile Ad Hoc Networks

G. Satyachellayi*, T. Veerraju**
*M.Tech Student, CSE Department, Sri AdityaEngineering College, A.P, India
**Associate professor, CSE Department, Sri AdityaEngineering College, A.P, India

Abstract

There are several services that build on the availability of closest node location information like geographic routing in spontaneous networks, data gathering in sensor networks, movement coordination among autonomous robotic nodes, location specific services for hand held devices and danger warning or traffic monitoring in vehicular networks. Ad hoc networking protocols and location-aware services require that mobile nodes identify the location of their closest nodes. Such a process can be easily misuses or stop by opposed nodes. In absence of a priori trusted nodes, the spotting and identifying of closest node position presents challenges that have been scarcely investigated in the literature. Node can also send message from one to many nodes in a broadcasting manner here.

Index Terms: Closest Nodes, Mobile Ad Hoc Networks, Phantom Nodes, Position identification, Vehicular Networks.

I. INTRODUCTION

Location details has become an important asset in mobile systems, where a wide range of protocols and applications require knowledge of the position of the participating nodes. Geographic routing in impetuous networks, data gathering in sensor networks, movement coordination among autonomous robotic nodes, location-specific services for handheld devices, and danger warning or traffic monitoring in vehicular networks are all examples of services that build on the availability of closest node location information.

The correctness of node locations is therefore an all- important issue in mobile networks, and it becomes particularly challenging in the presence of adversaries aiming at harming the system. In these cases, we need solutions that let nodes 1) correctly establish their location in spite of attacks feeding false location information, and 2) verify the positions of their neighbors, so as to detect adversarial nodes announcing false locations.

This system focus on the latter aspect, hereinafter referred to as closest node location identification (NPV for short). Specifically, we deal with a mobile ad hoc network, where a pervasive infrastructure is not present, and the location data must be obtained through node-to-node communication. Such a scenario is of particular interest since it leaves the door open for adversarial nodes to misuse or stop the location-based services.

For example, by advertising forged positions, adversaries could bias geographic routing or data gathering processes, attracting network traffic and then eavesdropping or discarding it. Similarly, counterfeit positions could grant adversaries unauthorized access to location-dependent services, let vehicles forfeit road tolls, stop vehicular traffic or endanger passengers and drivers.

A fully distributed, lightweight NPV procedure that enables each node to acquire the locations advertised by its neighbors, and assess their truthfulness. We therefore propose an NPV protocol that has the following features:

. It is designed for impetuous ad hoc environments, and, as such, it does not rely on the presence of a trusted infrastructure or of a priori trustworthy nodes;
. It leverages cooperation but allows a node to perform all identification procedures autonomously. This approach has no need for lengthy interactions, e.g., to reach a consensus among multiple nodes, making our scheme suitable for both low- and high- mobility environments;
. It is reactive, meaning that it can be executed by any node, at any point in time, without prior knowledge of the neighborhood;
. It is robust against independent and colluding adversaries;
. It is lightweight, as it generates low overhead traffic.

Additionally, our NPV scheme is compatible with state-of- the-art security architectures, including the ones that have been proposed for vehicular networks [1], [2], which represent a likely deployment environment for NPV.The rest of the project is organized as follows:In Section2, we review previous works, highlighting the novelty of our solution. In Section 3, we describe the system model, while the communication protocol, the objectives of the identification procedure and our main results are outlined in Section 4. The details of the NPV protocol and of identification tests are then presented in Section5, Finally, we provide a performance evaluation of the protocol in a vehicular scenario in Section 6, and draw conclusions in Section.

2.RELATEDWORK

Although the literature carries a multitude of ad hoc security protocols addressing a number of problems related to NPV, there are no lightweight, robust solutions to NPV that can operate autonomously in an open, ephemeral environment, without relying on trusted nodes. Below, we list relevant works and highlight the novelty of our contribution. For clarity of presentation, we first review solutions to some NPV-related problems, such as secure positioning and secure discovery, and then we discuss solutions specifically addressing NPV. Securely determining own location. In mobile environments, self-localization is mainly achieved through Global Navigation Satellite Systems, e.g., GPS, whose security can be provided by cryptographic and no cryptographic defense mechanisms [3]. Alternatively, terrestrial special- purpose infrastructure could be used [4], [5], along with techniques to deal with no honest beacons [6]. We remark that this problem is orthogonal to the problem of NPV. In the rest of this project, we will assume that devices employ one of the techniques above to securely determine their own position and time reference.

Secure neighbor discovery (SND) deals with the identification of nodes with which a communication link can be established or that are within a given distance. SND is only a step toward the solution we are after: simply put, an adversarial node could be securely discovered as neighbor and be indeed a neighbor (within some SND range), but it could still cheat about its position within the same range.

In other words, SND is a subset of the NPV problem, since it lets a node assess whether another node is an actual neighbor but it does not verify the location it claims to be at. SND is most often employed to counter wormhole attacks [7], [8]; practical solutions to the SND problem have been proposed in [9], while properties of SND protocols with proven secure solutions can be found in [10], [11].

Neighbor position identification was studied in the context of ad hoc and sensor networks; however, existing NPV schemes often rely on fixed [12], [13] or mobile [14] trustworthy nodes, which are assumed to be always available for the identification of the positions announced by third parties. In ad hoc environments, however, the pervasive presence of either infrastructure or neighbor nodes that can be aprioristically trusted is quite unrealistic. Thus, we devise a protocol that is autonomous and does not require trustworthy neighbors.

In [15], an NPV protocol is proposed that first lets nodes calculate distances to all neighbors, and then commends that all triplets of nodes encircling a pair of other nodes act as verifiers of the pair's positions. This scheme does not rely on trustworthy nodes, but it is designed for static sensor networks, and requires lengthy multiround computations involving several nodes that seek consensus on a common neighbor identification. Furthermore, the resilience of the protocol in [15] to colluding attackers has not been demonstrated. The scheme in [16] suits static sensor networks too, and it requires several nodes to exchange information on the signal emitted by the node whose location has to be verified. Moreover, it aims at assessing not the position but whether the node is within a given region or not. Our NPV solution, instead, allows any node to validate the position of all of its neighbors through a fast, one-time message exchange, which makes it suitable to both static and mobile environments. Additionally, we show that our NPV scheme is robust against several different colluding attacks. Similar differences can be found between our work and [17].

In [18], the authors propose an NPV protocol that allows nodes to validate the position of their neighbors through local observations only. This is performed by checking whether subsequent positions announced by one neighbor draw a movement over time that is physically possible. The approach in [18] forces a node to collect several data on its neighbor movements before a decision can be taken, making the solution unfit to situations where the location information is to be obtained and verified in a short time span. Moreover, an adversary can fool the protocol by simply announcing false positions that follow a realistic mobility pattern. Conversely, by exploiting cooperation among nodes, our NPV protocol is 1) reactive, as it can be executed at any instant by any node, returning a result in a short time span, and 2) robust to fake, yet realistic, mobility patterns announced by adversarial nodes over time.

To our knowledge, our protocol is the first to provide a fully distributed, lightweight solution to the NPV problem that does not require any infrastructure or a priori trusted neighbors and is robust to several different attacks, including coordinated attacks by colluding adversaries. Also, unlike previous works, our solution is suitable for both low and high mobile environments and it only assumes RF communication. Indeed, non-RF communication, e.g., infrared or ultrasound, is unfeasible in mobile networks, where non-line-of-sight conditions are frequent and device-to- device distances can be in the order of tens or hundreds of meters. An early version of this work, sketching the NPV protocol and some of the identification tests to detect independent adversaries, can be found in [19].

3.SYSTEM AND ADVERSARY MODEL

We consider a mobile network and define as communication neighbors of a node all the other nodes that it can reach directly with its transmissions . We assume that each node knows its own position with some maximum error p , and that it shares a common time reference with the other nodes: both requirements can be met by equipping communication nodes with GPS receivers.

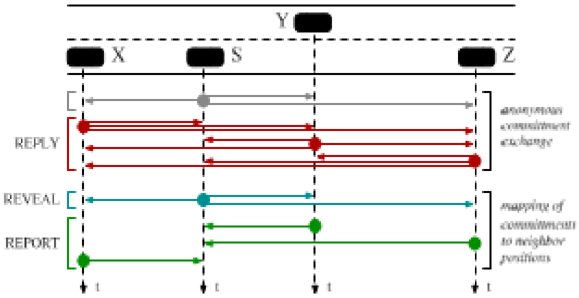

Y
X
S
Z
REPLY
REVEAL
REPORT
anonymous commitment exchange
mapping of commitments to neighbor positions
t
t
t
t

Fig. 1. Message exchange overview, during one instance of the NPV protocol.

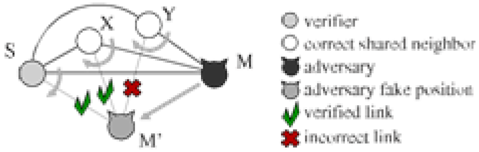
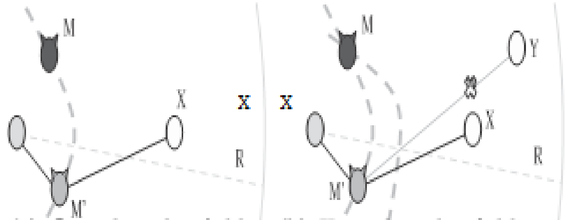


Fig. 2. Example of topological information stored by verifier S at the end of the message exchange and effect of a fake position announcement by M .

nodes can perform Time-of-Flight-based RF ranging with a maximum error equal to r . As discussed in [15], this is a reasonable assumption, although it requires modifications to off-the-shelf radio interfaces; also, promising techniques for precise ToF-based RF ranging have been developed [20].

We assume that node positions do not vary significantly during a protocol execution, since a complete message exchange takes no more than a few hundreds of milli- seconds. The relative spatial movements of the nodes during such a period are taken into account through the tolerance value m .

Nodes carry a unique identity2 and can authenticate messages of other nodes through public key cryptography. In particular, we assume that each node X owns a private key, kX , and a public key, KX , as well as a set of one-time use keys {k0 ; K0 }, as proposed in emerging architectures for secure and privacy-enhancing communication [2]. Node X can encrypt and decrypt data with its keys and the public keys of other nodes; also, it can produce digital signatures (SigX) with its private key. We assume that the binding between X and KX can be validated by any node, as in state-of-the-art secure communication architectures [2].



(a) One shared neighbor (b) Two shared neighbors

Nodes are correct if they comply with the NPV protocol, and adversarial if they deviate from it. As authentication essentially thwarts external adversaries, we focus on the more powerful internal ones, i.e., nodes that possess the cryptographic material to participate in the NPV and try to exploit it, by advertising arbitrarily erroneous own posi- tions or inject misleading information. Internal adversaries cannot forge messages on behalf of other nodes whose keys they do not have. Thus, attacks against the cryptosystem are not considered, as correct implementation of cryptographic primitives makes them computationally infeasible.

We further classify adversaries into: knowledgeable, if at each time instant they know positions and (temporary) identities of all their communication neighbors, and unknowledgeable, otherwise; independent, if they act indivi- dually, and colluding, if they coordinate their actions.

The protocol as well as the gist of its resilience analysis. Detailed discussions of message format, verification test is provided in Section 5.

A verifier, S, can initiate the protocol at any time instant, by triggering the 4-step message exchange depicted in Fig. 1, within its 1-hop neighborhood. The aim of the message exchange is to let S collect information it can use to compute distances between any pair of its communication neighbors. To that end, POLL and REPLY messages are first broadcasted by S and its neighbors, respectively.

These messages are anonymous and take advantage of the broadcast nature of the wireless medium, allowing nodes to record reciprocal timing information without disclosing their identities. Then, after a REVEAL broadcast by the verifier, nodes disclose to S, through secure and authenti- cated REPORT messages, their identities as well as the anonymous timing information they collected. The verifier S uses such data to match timings and identities; then, it uses the timings to perform ToF-based ranging and compute distances between all pairs of communicating

4.COOPERATIVE NPV: AN OVERVIEW

We propose a fully distributed cooperative scheme for NPV, which enables a node, hereinafter called the verifier, to discover and verify the position of its communication neighbors. For clarity, here we summarize the principles of nodes in its neighborhood.

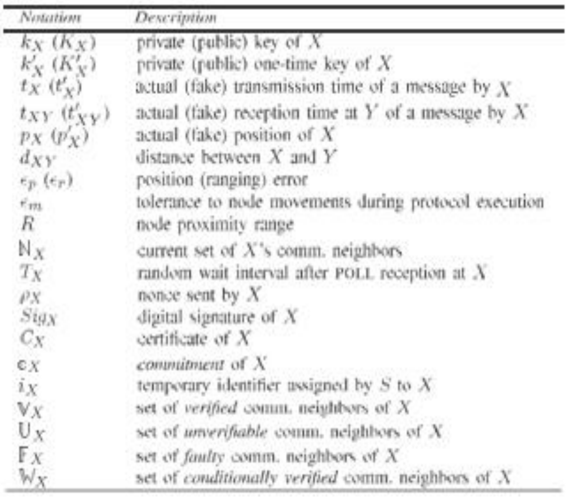
Once S has derived such distances, it runs several position verification tests in order to classify each candidate neighbor as either:

1. Verified, i.e., a node the verifier deems to be at the claimed position;
2. Faulty, i.e., a node the verifier deems to have announced an incorrect position;
3. Unverifiable, i.e., a node the verifier cannot prove to be either correct or faulty, due to insufficient information.

Clearly, the verification tests aim at avoiding false negatives (i.e., adversaries announcing fake positions that are deemed verified) and false positives (i.e., correct nodes whose positions are deemed faulty), as well as at minimizing the number of unverifiable nodes. We remark that our NPV scheme does not target the creation of a consistent "map" of neighborhood relations throughout an ephemeral network: rather, it allows the verifier to independently classify its neighbors.

The basic principle the verification tests build upon is best explained by means of the example in Fig. 2. There, M is a malicious node announcing a false location M 0 , so as to fraudulently gain some advantage over other nodes. The figure portrays the actual network topology with black edges, while the modified topology, induced by the fake position announced by M , is shown with gray edges. It is evident that the displacement of M to M 0 causes its edges with the other nodes to rotate e. The tests thus look for discrepancies in the node distance information to identify incorrect node positions.

TABLE 1
Summary of Notations



Notation	*Description*
k_X (K_X)	private (public) key of X
k'_X (K'_X)	private (public) one-time key of X
t_X (t'_X)	actual (fake) transmission time of a message by X
t_{XY} (t'_{XY})	actual (fake) reception time at Y of a message by X
p_X (p'_X)	actual (fake) position of X
d_{XY}	distance between X and Y
ϵ_p (ϵ_r)	position (ranging) error
ϵ_m	tolerance to node movements during protocol execution
R	node proximity range
$\mathbb{N}_X$	current set of X's comm. neighbors
T_X	random wait interval after POLL reception at X
ρ_X	nonce sent by X
Sig_X	digital signature of X
C_X	certificate of X
$\mathfrak{c}_X$	*commitment* of X
i_X	temporary identifier assigned by S to X
$\mathbb{V}_X$	set of *verified* comm. neighbors of X
$\mathbb{U}_X$	set of *unverifiable* comm. neighbors of X
$\mathbb{F}_X$	set of *faulty* comm. neighbors of X
$\mathbb{W}_X$	set of *conditionally verified* comm. neighbors of X

A malicious node, knowing the protocol, can try to outsmart the tests in a number of different ways. Section 6 contains a comprehensive discussion of the protocol resilience, covering conceivable attack strategies that adver- sarial nodes could adopt. Overall, our analysis proves that:

. An unknowledgeable adversary has no possibility of success against our NPV protocol;

. An independent knowledgeable adversary M can move at most two links (with the verifier S and with a shared neighbor X) without being detected: how- ever, any additional link (e.g., with another shared neighbor Y) leads to inconsistencies between dis- tances and positions that allow to identify the attacker: this is the situation depicted in Fig. 2. In a nutshell, independent adversaries, although knowl- edgeable, cannot harm the system;

. Colluding knowledgeable adversaries can announce timing information that reciprocally validate their distances, and pose a more dangerous threat to the system. However, we prove that an overwhelming presence of colluders in the verifier neighborhood is required for an attack to be successful. Additionally, simulations in realistic scenarios prove the robust- ness of the NPV protocol even against large groups of colluding knowledgeable adversaries.

5.NPV PROTOCOL

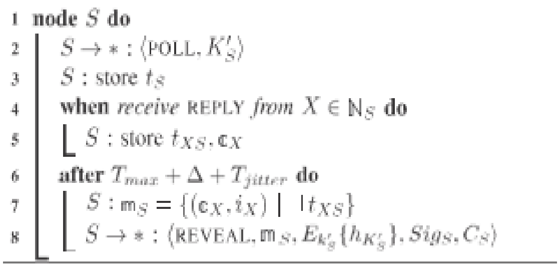
We detail the message exchange between the verifier and its communication neighbors, followed by a description of the tests run by the verifier. Table 1 summarizes the notations used throughout the protocol description.

5.1 Protocol Message Exchange

The value pX is the current position of X, and INX is the current set of its communication neighbors. We denote by tX the time at which a node X starts a broadcast transmission and by tXY the time at which a node Y starts receiving it. Note that these time values refer to the actual instant at which the node starts transmitting/receiving the first bit of the message at the physical layer. To retrieve the exact transmission and reception time instants, avoiding the unpredictable latencies introduced by interrupts trig- gered at the drivers level, a solution such as that implemented in is required.3 Furthermore, the GPS receiver should be integrated in the 802.11 card; software defined radio solutions combining GPS and 802.11 capabil- ities are proposed.

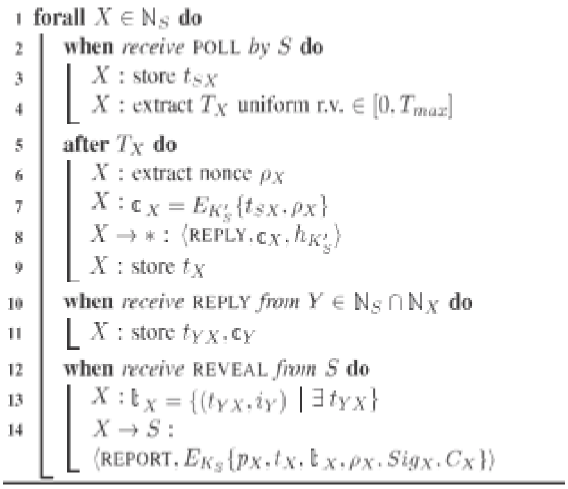
Now, consider a verifier S that initiates the NPV protocol. The message exchange procedure is outlined in Algorithm 1 for S, and in Algorithm 2 for any of S s communication neighbors.

Algorithm 1. Message exchange protocol: verifier.



```
1 node $S$ do
2     $S \rightarrow * : \langle \text{POLL}, K'_S \rangle$
3     $S$ : store $t_S$
4     when receive REPLY from $X \in \mathbb{N}_S$ do
5         $S$ : store $t_{XS}, \mathbb{c}_X$
6     after $T_{max} + \Delta + T_{jitter}$ do
7         $S : \mathbb{m}_S = \{(\mathbb{c}_X, i_X) \mid \, |t_{XS}\}$
8         $S \rightarrow * : \langle \text{REVEAL}, \mathbb{m}_S, E_{k'_S}\{h_{K'_S}\}, Sig_S, C_S \rangle$
```

Algorithm2. Message exchange protocol: any neighbor.



```
1 forall $X \in \mathbb{N}_S$ do
2     when receive POLL by $S$ do
3         $X$ : store $t_{SX}$
4         $X$ : extract $T_X$ uniform r.v. $\in [0, T_{max}]$
5     after $T_X$ do
6         $X$ : extract nonce $\rho_X$
7         $X : \mathbb{c}_X = E_{K'_S}\{t_{SX}, \rho_X\}$
8         $X \rightarrow * : \langle \text{REPLY}, \mathbb{c}_X, h_{K'_S} \rangle$
9         $X$ : store $t_X$
10    when receive REPLY from $Y \in \mathbb{N}_S \cap \mathbb{N}_X$ do
11        $X$ : store $t_{YX}, \mathbb{c}_Y$
12    when receive REVEAL from $S$ do
13        $X : \mathbb{t}_X = \{(t_{YX}, i_Y) \mid \exists t_{YX}\}$
14        $X \rightarrow S$ :
          $\langle \text{REPORT}, E_{K_S}\{p_X, t_X, \mathbb{t}_X, \rho_X, Sig_X, C_X\} \rangle$
```

POLL message. The verifier starts the protocol by broadcasting a POLL whose transmission time tS it stores locally (Algorithm 1, lines 2-3). The POLL is anonymous, since 1) it does not carry the identity of the verifier, 2) it is transmitted employing a fresh, software-generated MAC address, and 3) it contains a public key K^{0s} taken from S's pool of anonymous one-time use keys that do not allow neighbors to map the key onto a specific node. We stress that keeping the identity of the verifier hidden is important in order to make our NPV robust to attacks . Since a source address has to be included in the MAC-layer header of the message, a fresh, software-generated MAC address is needed; note that this is considered a part of emerging cooperative systems [2]. Including a one-time key in the POLL also ensures that the message is fresh (i.e., the key acts as a nonce).

3. This leads to a timing precision of around 23 ns, dictated by the 44 MHz clock of standard 802.11a/b/g cards. As mentioned above, we account for these errors through the r parameter.

REPLY message. A communication neighbor X 2 INS that receives the POLL stores its reception time tSX , and extracts a random wait interval TX 2 ½0; Tmax (Algorithm 2, lines 2-4). After TX has elapsed, X broadcasts an anon- ymous REPLY message using a fresh MAC address, and locally records its transmission time tX (Algorithm 2, lines 5-9). For implementation feasibility, the physical layer transmission time cannot be stamped on the REPLY, but it is stored by X for later use. The REPLY contains some information encrypted with S s public key (K0), specifically the POLL reception time and a nonce X used to tie the REPLY to the next message sent by X: we refer to these data as X's commitment, Cj X (Algorithm 2, line 7). The hash hK0 , derived from the public key of the verifier, K0 , is also included to bind POLL and REPLY belonging to the same message exchange.

Upon reception of a REPLY from a neighbor X, the verifier S stores the reception time tXS and the commitment Cj X (Algorithm 1, lines 4-5). When a different neighbor of S, e.g., Y , Y 2 INS \ INX , broadcasts a REPLY too, X stores the reception time tYX and the commitment Cj Y (Algorithm 2, lines 10-11). Since REPLY messages are anonymous, a node records all commitments it receives without knowing their originators.

REVEAL message. After a time Tmax þ þ Tjitter , the verifier broadcasts a REVEAL message using its real MAC address (Algorithm 1, line 6). accounts for the propaga- tion and contention lag of REPLY messages scheduled at time Tmax , and Tjitter is a random time added to thwart jamming efforts on this message. The REVEAL contains: 1) a map ImS , that associates each commitment Cj X received by the verifier to a temporary identifier iX (Algorithm 1, line 7);

2) a proof that S is the author of the original POLL through the encrypted hash Ek0 fhK0 g; 3) the verifier identity, i.e., its

certified public key and signature (Algorithm 1, line 8).

Note that using certified keys curtails continuous attempts at running the protocol by an adversary who aims at learning neighbor positions (i.e., at becoming knowledge- able) or at launching a clogging attack (see Section 6.4).

REPORT message. Once the REPORT message is broad-cast and the identity of the verifier is known, each neighbor X that previously received S's POLL unicasts to S an encrypted, signed REPORT message. The REPORT carries X's position, the transmission time of X's REPLY, and the list of pairs of reception times and temporary identifiers referring to the REPLY broadcasts X received (Fig. 2, lines 12-14). The identifiers are obtained from the map ImS included in the REVEAL message. Also, X discloses its own identity by including in the message its digital signature and certified public key; through the nonce X , it correlates the REPORT to its previously issued REPLY. We remark that all sensitive data are encrypted using S s public key, KS , so that eavesdropping on the wireless channel is not possible. At the end of the message exchange, only the verifier knows all positions and timing information. If needed, certified keys in REPORT messages allow the matching of such data and node identities (temporary or long-term, with the help of an authority if needed [2]).

5.2 Position Verification

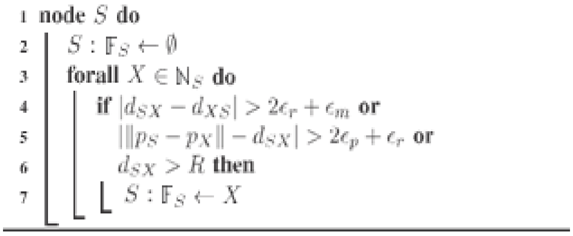
Once the message exchange is concluded, S can decrypt the received data and acquire the position of all neighbors that participated in the protocol, i.e., fpX ; 8X 2 INS g. The verifier S also knows the transmission time tS of its POLL and learns that of all subsequent REPLY messages, i.e., ftX ; 8X 2 INS gas well as the corresponding reception times recorded by the recipients of such broadcasts, i.e., ftXY; 8X; Y 2 INS [fSgg. Applying a ToF-based technique, S thus computes its distance from each communication neighbor, as well as the distances between all neighbor pairs sharing a link. More precisely, by denoting with c the speed of light, the verifier computes, for any communicating pair ðX; Y Þ with X; Y 2 INS [fSg, two distances: dXY ¼ ðtXY tX Þ c, from the timing informa- tion related to the broadcast message sent by X, and dYX ¼ ðtYX tY Þ c, from the information related to the broadcast message by Y .

Once such distances have been computed, S can run the following three verification tests to fill the sets IFS, jVS, and UUS with, respectively, faulty, verified and unverifiable nodes.

5.2.1 The Direct Symmetry Test (DST)

DST is the first verification performed by S and is detailed in Algorithm 3. There, j j denotes the absolute value operator and kpX pY k the euclidean distance between movements during the protocol execution. The second check verifies that the position advertised by the neighbor is consistent with such distances, within an error margin of 2 p þ r (Algorithm 3, line 5). Although trivial, this check is fundamental since it correlates positions to computed distances: without it, an attacker could fool the verifier by simply advertising an arbitrary position along with correct broadcast transmission and reception timings. Finally, as a sanity check, S verifies that dSX is not larger than R (Algorithm 3, line 6). The verifier tags a neighbor as faulty if a mismatch is found in any of these checks,4 since this implies an inconsistency between the position pX and the timings announced by the neighbor (tSX , tX) or recorded by the verifier (tXS , tS).

Algorithm 3. Direct Symmetry Test (DST)



```
1 node $S$ do
2     $S : \mathbb{F}_S \leftarrow \emptyset$
3     forall $X \in \mathbb{N}_S$ do
4         if $|d_{SX} - d_{XS}| > 2\epsilon_r + \epsilon_m$ or
5            $|\|p_S - p_X\| - d_{SX}| > 2\epsilon_p + \epsilon_r$ or
6            $d_{SX} > R$ then
7             $S : \mathbb{F}_S \leftarrow X$
```

6. PERFORMANCE EVALUATION

We evaluated the performance of our NPV protocol in a vehicular scenario. Results obtained in a pedestrian scenario are available as supplemental material, which can be found on the Computer Society Digital Library at http:// doi.ieeecom putersociety.org/10.1109/TMC. 2011.258.

We focus on knowledgeable adversaries whose goal is to make the verifier believe their fake positions, and we describe the best attack strategy they can adopt in Section 7.1. Such a strategy, will be assumed while deriving the results shown in Section 7.2.

The results, which therefore represent a worst case analysis of the proposed NPV, are shown in terms of the probability that the tests return false positives and false negatives as well as of the probability that a (correct or adversary) node is tagged as unverifiable. In addition, we plot the average difference between the true position of a successful adversary and the fake position it advertises, as well as the overhead introduced by

4. The latter two checks are performed on both dSX and dXS , however in Algorithm 3 they are done on dSX only, for clarity of presentation.

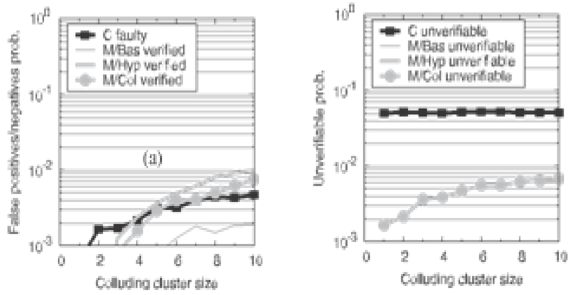
our NPV scheme. The results on attacks aimed at discrediting the position of other nodes are omitted, since they are very close to those we present later in this section.

6.1 Results

We employed movement traces representing vehicle traffic over a real-world road topology. More precisely, we considered car movements within a 20 km2 portion of the Karlsruhe urban area depicted in Fig. 8, extracting 3 hours of vehicular mobility that reproduce mild to heavy traffic density conditions. These synthetic traces were generated using the IDM-LC model of the VanetMobiSim simulator, which takes into account car-to-car interactions, traffic lights, stop signs, and lane changes, and has been proven to realistically reproduce vehicular movement patters in urban scenarios [31].

In our simulations, we set Tmax ¼ 200 ms, Tjitter ¼ 50 ms, ¼ 1 ms and assume that CSMA/CA is used to access the wireless medium, hence messages can be lost due to collisions. Unless otherwise specified, we fix the proximity range, R, which is equal to the maximum nominal transmis- sion range, to 250 m (resulting in an average neighborhood size of 73.4 nodes), while r ¼ 6:8 m, p ¼ 10 m, and the tolerance value m ¼ 5 m (roughly corresponding to the case of two vehicles moving at 50 km/h in opposite directions).

To evaluate the performance of our NPV, at every simulation second we randomly select 1 percent of the nodes as verifiers. Then, for each verifier, we compare the outcome of the verification tests with the actual nature of the neighbors. We consider colluding adversaries acting in groups, referred to as clusters. Note that a colluding cluster size equal to 1 corresponds to independent attacks. Also, adversaries are knowledgeable, i.e., they perfectly know the identity and location of all colluding and noncolluding neighbors, and always adopt the best attack strategy as described in Section 7.1. In the following, unless otherwise specified, adversaries amount to 5 percent of the overall nodes and are divided into clusters of five colluders each.



In the legend of the plots, C stands for correct node (e.g., the label "C faulty" refers to the probability of false positives), while M/Bas, M/Hyp, and M/Col stand for adversaries launching, respectively, the basic, hyperbola-based and collinear attack (e.g., the label "M/Bas verified" refers to the probability of false negatives due to basic attacks).

We first examine the NPV protocol performance for different values of colluding cluster sizes and R ¼ 250 m (Figs. 9a and 9b).

The false negative/positive probability in Fig. 9a clearly shows that1) the chance of wrong classification reaches 0.01 only for a very large adversarial cluster size, namely 10, 2) the hyperbola-based and the collinear attacks are the most threatening and 3) an attack by the colluders is most effective in passing themselves off as verified when there are at least three of them. The cluster size also affects the colluders ability to disrupt the positioning of correct nodes, which exhibit as high as a 0.4 percent chance to be tagged as faulty.

Conversely, as shown in Fig. 9b, the cluster size does not cause more correct nodes to be unverifiable, since the main reason for correct nodes to be tagged as unverifiable is the lack of noncollinear neighbors that can verify them. The chance for an adversary to be unverifiable increases with the cluster size, although it is significant only in case of collinear attacks. This is in agreement with the fact that the outcome of the collinear attack is the avoidance of a sizable number of cross-checks between the adversary and correct nodes, thus likely leading the adversary to be tagged as unverifiable.

The neighborhood size proves to play an important role, as evident in Figs. 9c and 9d where we consider a 5-colluder cluster and vary the transmission range. A small R (hence few neighbors) affects the NPV capability to correctly tag a node. Widening the transmission range with a fixed colluding cluster size significantly favors the verifier, allowing it to reach a conclusive and exact verdict on either correct or adversary nodes: the larger the R, the higher the number of cross-checks involving correct nodes in the CST. We note that, for transmission ranges larger than 300 m, we obtain false positive/negative probabilities that are smaller than 0.001. Below 150-m ranges (corresponding to an average neighborhood size of 12 nodes), such probabilities are still 0.01.

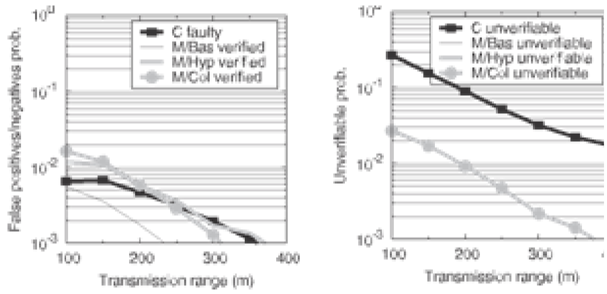


Fig. 9. Probability that a neighbor is tagged incorrectly or as unverifiable, versus the colluder cluster size (a,b), and versus R (c,d). C: correct; M/Bas,/Hyp, and M/Col: adversaries launching the basic, hyperbola-based and collinear attack, each combined with the REPLY-disregard attack.

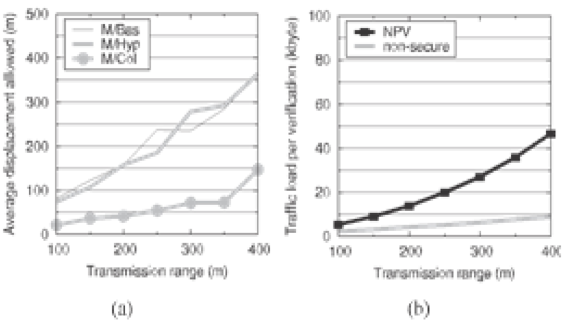


Fig. 10. Displacement gain of adversaries running a successful attack against the NPV (a) and traffic load induced by one instance of the protocol (b).

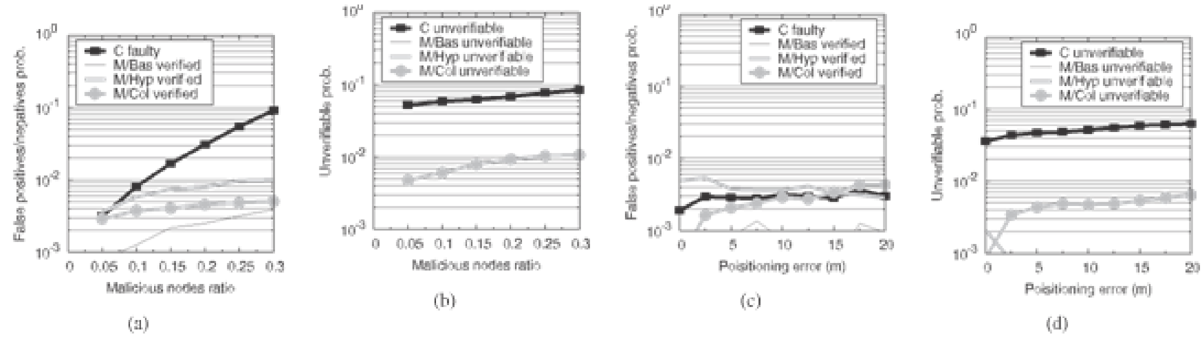


Fig. 11. Probability that a neighbor is tagged incorrectly or as unverifiable, versus the ratio of adversaries (a,b), and position error (c,d). C: correct; M/Bas, M/Hyp, and M/Col: adversaries launching the basic, hyperbola-based, and collinear attack, each combined with the REPLY-disregard attack.

Beside the impact of the cluster size and of the transmission range, it is important to understand the effect of the percentage of adversaries in the vehicular network. Thus, in Fig. 11a we fix R to 250 m and the cluster size to 5, and we show the robustness of our NPV to the density of adversaries: the probability that adversaries are verified increases ever so slightly with their density. The highest effect is on the probability of correct nodes being tagged as faulty, which however reaches its highest value (0.1) only for 30 percent of adversaries in the network. A further effect of the growing presence of adversaries, as shown in Fig. 11b, is the unverifiable tag being slapped onto more correct nodes. A final observation can be made looking at the false positive/negative probability as the positioning error varies (Figs. 11c and 10d). Interestingly, for any positioning error different from 0, the metrics are only marginally affected.

Finally, we further increase the level of detail of our analysis and study the advantage obtained by adversaries that perform a successful attack against the NPV protocol. Such an adversarial gain is expressed in terms of spatial displacement, i.e., difference of position between the real and fraudulently advertised locations of the successful attacker: clearly, a larger displacement range implies a higher freedom of movement, which, in turn, enables potentially more dangerous actions against the system. The results in Fig. 11a are broken down based on the type of attack launched by the successful adversary, and are limited to the impact of the transmission range, since the other parameters did not show significant influence on the displacement of successful attackers.

We can observe that successful collinear attacks yield small advantage for adversaries, who are forced to announce positions quite close to their real locations. Moreover, we recall that these attacks constrain adversaries to advertise fake positions along a precise axis, thus further limiting their freedom of movement. We can conclude that collinear attacks, typically those with the highest chances of success as previously discussed, are also those resulting in the smallest gain for the adversaries. Conversely, basic attacks allow the largest average displacements, but we showed that they have extremely low success probability. The hyperbola-based attacks appear then to be the most dangerous ones, if the displacement gain is taken into consideration. However, such a gain becomes significant only for large transmission ranges, in presence of which we already observed that the actual success probability of the attacks becomes negligible.

Finally, we comment on the overhead introduced by our scheme. The NPV protocol generates at most 2n þ 2 messages for one execution initiated by a verifier with n communication neighbors. Also, NPV messages are relatively small in size: with SHA-1 hashing and ECDSA-160 encryption , the length of signatures is 21 bytes (with coordinates compression). Assuming that messages include headers with 4-byte source and destination identifiers and 1-byte message type field, POLL, REPLY, and REVEAL are 26, 71, and 67 bytes in size, respectively. The REPORT length depends on the quantity of common neighbor data it carries, amounting to 4 bytes per shared neighbor: information on more than 360 neighbors can thus fit in a single IP packet.

Fig. 11b portrays the traffic induced on the network by one instance of the NPV protocol. The plot only accounts for transmission range variations since, once more, the other parameters do not have an impact on the overhead. We can observe that security comes at a cost, since the traffic load of the NPV protocol is higher than that of a basic nonsecure neighbor position discovery, consisting of only one poll and associated position replies from neighbors. More precisely, the NPV protocol overhead is comparable to that of the nonsecure discovery for smaller transmission ranges, while the difference tends to increase for larger ranges. However, the cost of the NPV protocol is affordable in absolute terms, since one run requires just a few tens of kbytes to be exchanged among nodes, even in presence of dense networks and large transmission ranges. Note that the results above do not take into account the overhead induced by the distribution of certificates, as it is out of the scope of this work (the interested reader can refer to [26]).

Summary. Given that we assumed the best possible conditions for the adversaries, the above results prove our NPV to be highly resilient to attacks. Indeed, we observed typical probabilities of false positives/negatives below
1 percent, while that of a node being tagged as unverifiable is below 5 percent. Moreover, we showed that a significant portion of the successful attacks yields small advantage to the adversaries in terms of displacement. Finally, the overhead introduced by the NPV protocol is reasonable, as it does not exceed a few tens of kbytes even in the most critical conditions.

7.CONCLUSION

We presented a distributed solution for closest node identification, which allows any node in a mobile ad hoc network to verify the position of its communication neighbors without relying on a priori trustworthy nodes. Our analysis showed that our protocol is very robust to attacks by independent as well as colluding adversaries, even when they have perfect knowledge of the neighborhood of the verifier with the concept of phantom node, closest node identification. Simulation results confirm that our solution is effective in identifying nodes advertising false positions, while keeping the probability of false positives low. Only an overwhelming presence of colluding adversaries in the neighborhood of the verifier, or the unlikely presence of fully collinear network topologies, can degrade the effectiveness of our NPV. Here message can be broadcast to closest nodes. Future work will aim at integrating the NPV protocol in higher layer protocols, as well as at extending it to a proactive paradigm, useful in presence of applications that need each node to constantly verify the position of its neighbors.

REFERENCES

[1] 1609.2-2006: IEEE *Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages*, IEEE, 2006.

[2] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "*Secure Vehicular Communications: Design and Architecture,*" IEEE Comm. Magazine, vol. 46, no. 11, pp. 100-109, Nov. 2008.

[3] P. Papadimitratos and A. Jovanovic, "*GNSS-Based Positioning: Attacks and Countermeasures*," Proc. IEEE Military Comm. Conf. (MILCOM), Nov. 2008.

[4] L. Lazos and R. Poovendran, "*HiRLoc: High-Resolution Robust Localization for Wireless Sensor Networks*," IEEE J. Selected Areas in Comm., vol. 24, no. 2, pp. 233-246, Feb. 2006.

[5] R. Poovendran and L. Lazos, "*A Graph Theoretic Framework for Preventing the Wormhole Attack,*" Wireless Networks, vol. 13, pp. 27-59, 2007.

[6] S. Zhong, M. Jadliwala, S. Upadhyaya, and C. Qiao, "*Towards a Theory of Robust Localization against Malicious Beacon Nodes,*" Proc. IEEE INFOCOM, Apr. 2008.

[7] J. Eriksson, S. Krishnamurthy, and M. Faloutsos, "*TrueLink: Practical Countermeasure to the Wormhole Attack in Wireless Networks,"* Proc. IEEE 14th

Int'l Conf. Network Protocols (ICNP), Nov. 2006.

[8] R. Maheshwari, J. Gao, and S. Das, "*Detecting Wormhole Attacks in Wireless Networks Using Connectivity Information*," Proc. IEEE INFOCOM, Apr. 2007.

[9] R. Shokri, M. Poturalski, G. Ravot, P. Papadimitratos, and J.-P. Hubaux, "*A Practical Secure Neighbor Verification Protocol for Wireless Sensor Networks*," Proc. Second ACM Conf. Wireless Network Security (WiSec), Mar. 2009.

[10] M. Poturalski, P. Papadimitratos, and J.-P. Hubaux, "*Secure Neighbor Discovery in Wireless Networks: Formal Investigation of Possibility*," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), Mar. 2008.

[11] M. Poturalksi, P. Papadimitratos, and J.-P. Hubaux, "*Towards Provable Secure Neighbor Discovery in Wireless Networks*," Proc. Workshop Formal Methods in Security Eng., Oct. 2008.

[12] E. Ekici, S. Vural, J. McNair, and D. Al-Abri, "*Secure Probabilistic Location Verification in Randomly Deployed Wireless Sensor Networks,"* Elsevier Ad Hoc Networks, vol. 6, no. 2, pp. 195-209,2008.

[13] J. Chiang, J. Haas, and Y. Hu, "*Secure and Precise Location Verification Using Distance Bounding and Simultaneous Multi- lateration*," Proc. Second ACM Conf. Wireless Network Security (WiSec), Mar. 2009.

[14] S. C apkun, K. Rasmussen, M. Cagalj, and M. Srivastava, "*Secure Location Verification with Hidden and Mobile Base Stations,"* IEEE Trans. Mobile Computing, vol. 7, no. 4, pp. 470-483, Apr. 2008.

[15] S. C apkun and J.-P. Hubaux, "*Secure Positioning inWireless Networks*," IEEE J. Selected Areas in Comm., vol. 24, no. 2, pp. 221-232, Feb. 2006.

[16] A. Vora and M. Nesterenko, "*Secure Location Verification Using Radio Broadcast*," IEEE Trans. Dependable and Secure Computing, vol. 3, no. 4, pp. 377-385, Oct.-Dec. 2006.

[17] J. Hwang, T. He, and Y. Kim, "*Detecting Phantom Nodesin Wireless Sensor Networks,*" Proc. IEEE INFOCOM, May 2007.

[18] T. Leinmu¨ ller, C. Maiho¨ *fer, E. Schoch, and F. Kargl, "Improved Security in Geographic Ad Hoc Routing through Autonomous Position Verification,*" Proc. ACM Third Int'l Workshop Vehicular Ad Hoc Networks (VANET), Sept. 2006.

[19] J.-H. Song, V. Wong, and V. Leung, "*Secure Location Verification for Vehicular Ad-Hoc Networks*," Proc. IEEE Globecom, Dec. 2008.

[20] M. Fiore, C. Casetti, C.-F. Chiasserini, and P. Papadimitratos, "*Secure Neighbor Position Discovery in Vehicular Networks," Proc. IEEE/IFIP 10th Ann*. Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), June 2011.

[21] Fed. Highway Administration, "*High Accuracy-Nationwide Differential Global Positioning System Test and Analysis: Phase II Report,*" FHWA-HRT-05-034, July 2005.

BIOGRAPHIES

G. Satya Chellayi is Pursuing M.Tech in Computer Science from Sri Aditya Engineering College, Surampalem, A.P. Her area of interest includes Computer Networks, Mobile Communications, Data Base Management Systems, Data warehousing and Data Mining and Web Technologies.

Mr T.Veerraju, Asoc. Prof. Computer Science and Engineering at Sri Aditya Engineering College, Surampalem, E.G.Dt. His area of interest includes Computer Networks, Mobile Communications, Cryptography, Network Security, Data warehousing and Data Mining and Web Technologies.