Securing Mobile Ad-Hoc Network by Discovering and Verifying Neighborhood

Pranita D. Pandit[1], Prof. Pranjali Deshmukh[2]

[1, 2] P.R.Pote Collage of Engineering, SGBA University, Amravati, Maharashtra, India

Abstract: ***In a mobile ad hoc network without knowing neighbor node position which makes a chance to attackers to easily enter into the network. A growing number of ad hoc networking protocols and location-aware services require that mobile nodes learn the position of their neighbors. However, such a process can be easily abused or disrupted by adversarial nodes. In absence of a priori trusted nodes, the discovery and verification of neighbor positions presents challenges that have been scarcely investigated. Providing this protocol to a wireless ad hoc network makes it to be more secure. Results show that our protocol can determine attacks under the best possible conditions for the adversaries, with minimal false positive rates. Secure Neighbor Discovery which offers a measure of protection by allowing participating mobile nodes to securely determine if they are neighbors. Neighbor position verification designed for spontaneous ad hoc environments, and, as such, it does not rely on the presence of a trusted infrastructure or of a priori trustworthy nodes. The paper includes result of NPV protocols.***

Keywords: Mobile ad hoc networks, Neighbor discovery, Neighbor position verification.

1. Introduction

A mobile ad hoc network (**MANET**) is a self-configuring infrastructure-less network of mobile devices connected by wireless. It consists of a collection of mobile hosts that may communicate with each another from time to time. Due to mobility in MANETs, each device is free to move independently in any direction, and will therefore change its links to other devices frequently. The primary challenge in construction of a MANET is equipping each device to continuously maintain the information required to properly direct the traffic. Most traditional mobile ad hoc network routing protocols were designed focusing on the efficiency and performance of the network [9].

In [1], Location awareness has become an asset in mobile systems, where a wide range of protocols and applications require knowledge of the position of the participating nodes. 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, 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. The neighbor position verification (NPV) protocol is specifically; 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 disrupt the location-based services.

In this, design a powerful and secure neighbor verification protocol that adheres to the limited hardware capabilities of WSN, as it is demonstrated by implementation. In protocol, each node estimates its distance to the other nodes it can communicate with through a single hop. Then, nodes exchange information about their estimates. Next, a series of simple geometric tests is run by each node over the local neighborhood view it has obtained, in order to detect topology distortions created by wormhole attacks. Only those nodes that successfully pass the tests are verified to be actual communication neighbors.

2. Literature Review

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 [1]. For clarity of presentation, first review solutions to some NPV-related problems, such as secure positioning and secure discovery, and then discuss solutions specifically addressing NPV. Securely determining own location. In mobile environments, self-localization is mainly achieved through Global Navigation Satellite Systems, whose security can be provided by cryptographic and no cryptographic defense mechanisms [1].

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 [2]. SND is only a step toward the solution is 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 , practical solutions to the SND problem have been used in [3], while properties of SND protocols with proven secure solutions can be found in [4], [5].

Neighbor position verification was studied in the context of ad hoc and sensor networks; however, existing NPV schemes often rely on fixed [6], [7] or mobile trustworthy nodes, which are assumed to be always available for the verification of the positions announced by third parties. In ad hoc environment, the pervasive presence of either infrastructure or neighbor nodes that can be aprioristically trusted is quite unrealistic. Thus, design protocol that is autonomous and does not require trustworthy neighbors. In [7], an NPV protocol is 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 multi round computations involving several nodes that seek consensus on common neighbor verification. Furthermore, the resilience of the protocol in [3] to colluding attackers has not been demonstrated. To knowledge, protocol is the first to provide a fully distributed, lightweight solution to the NPV problem that does not require any infrastructure or a priori trust neighbors and is robust to several different attacks, including coordinated attacks by colluding adversaries.

3. Cooperative Npv: An Overview

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.

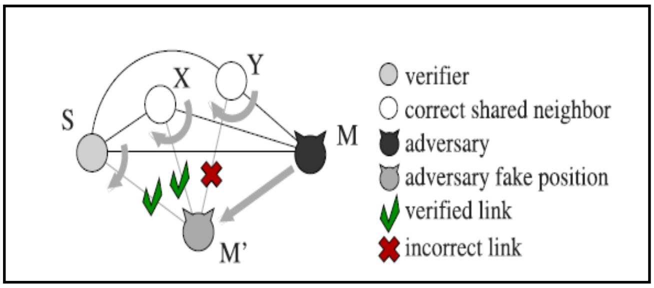


Figure 1: Example of topological information stored by verifier S at the end of the message exchange and effect of a fake position announcement by M [1].

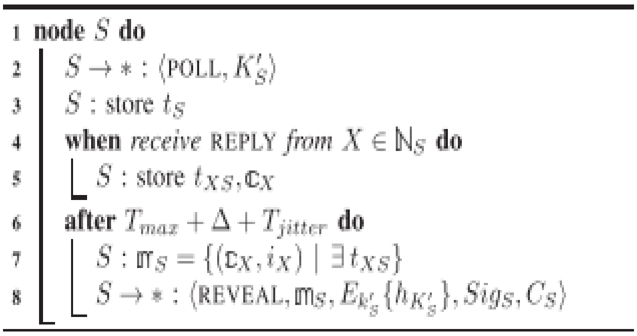
A verifier, S, can initiate the protocol at any time instant, by triggering the 4-step message exchange depicted, within its 1-hop neighborhood. The aim of the message exchange is to let S collect information it can use to compute distances between any pair of its communication neighbors. To that end,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

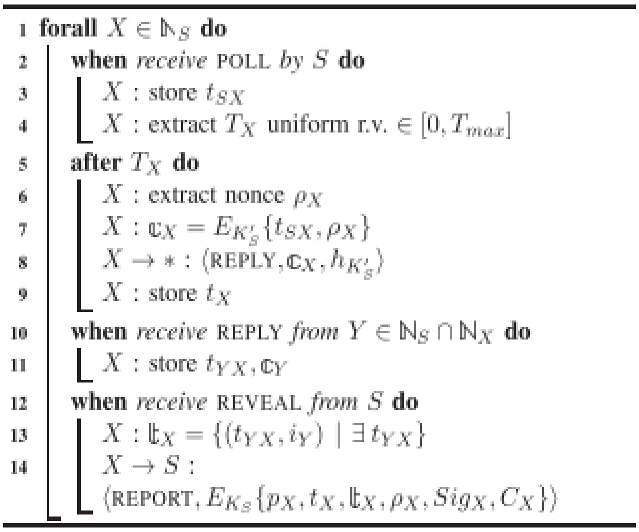
4. Implementation of Npv

For securing the basic technique are proposed. Firstly discover the secure neighborhood and then verify that neighborhood. Secure neighbor discovery deals with the identification of nodes with which a communication link can be established or that are within a given distance. The verification tests aim at avoiding false negatives and false positives as well as at minimizing the number of unverifiable nodes. The value p_X is the current position of X, and INX is the current set of its communication neighbors. Proposed system denote by t_X the time at which a node X starts a broadcast transmission and by t_{xy} 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. 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 [1].



```
node $S$ do
    $S \rightarrow * : \langle \text{POLL}, K'_S \rangle$
    $S$ : store $t_S$
    when receive REPLY from $X \in \mathbb{N}_S$ do
        $S$ : store $t_{XS}, \mathbb{C}_X$
    after $T_{max} + \Delta + T_{jitter}$ do
        $S : \mathbb{M}_S = \{(\mathbb{C}_X, i_X) \mid \exists t_{XS}\}$
        $S \rightarrow * : \langle \text{REVEAL}, \mathbb{M}_S, E_{K'_S}\{h_{K'_S}\}, Sig_S, C_S \rangle$
```

Algorithm 1: Message Exchange Protocol :Verifier[1]



```
forall $X \in \mathbb{N}_S$ do
    when receive POLL by $S$ do
        $X$ : store $t_{SX}$
        $X$ : extract $T_X$ uniform r.v. $\in [0, T_{max}]$
    after $T_X$ do
        $X$ : extract nonce $\rho_X$
        $X : \mathbb{C}_X = E_{K'_S}\{t_{SX}, \rho_X\}$
        $X \rightarrow * : \langle \text{REPLY}, \mathbb{C}_X, h_{K'_S} \rangle$
        $X$ : store $t_X$
    when receive REPLY from $Y \in \mathbb{N}_S \cap \mathbb{N}_X$ do
        $X$ : store $t_{YX}, \mathbb{C}_Y$
    when receive REVEAL from $S$ do
        $X : \mathbb{L}_X = \{(t_{YX}, i_Y) \mid \exists t_{YX}\}$
        $X \rightarrow S$ :
        $\langle \text{REPORT}, E_{K_S}\{p_X, t_X, \mathbb{L}_X, \rho_X, Sig_X, C_X\} \rangle$
```

Algorithm 2: Message Exchange protocol :Any Neighbor[1]

- **POLL message**
  A verifier S initiates this message. This message is anonymous. The verifier identity is kept hidden. Here software generated MAC addresses is used. This carries a public key K'S chosen from a pool of onetime use keys of S'.

- **REPLY message**
  A communication neighbor X receiving the POLL message will broadcast REPLY message after a time interval with a freshly generated MAC address. This also internally saves the transmission time. It contains some encrypted message with S public key (K'S).
- **REVEAL message**
  The REVEAL message broadcasting is done by using Verifier's real MAC address. It contains a map MS, a proof that S is the author of the original POLL and the verifier identity, i.e., its certified public key and signature.
- **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. The identifiers are obtained from the map MS included in the REVEAL message. Also, X discloses its own identity by including in the message its digital signature and certified public key.

5. Result Analysis

To evaluate the performance of our NPV, at every simulation second we randomly select 1% 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 focus on knowledgeable adversaries whose goal is to make the verifier believe their fake positions. Such a strategy, which depends on the neighborhood of the adversary and builds on a combination of the attacks, will be assumed while deriving the results shown. The results, which therefore represent 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 our evaluation, we compare the performances of NPV using Network Simulator 2.35 (NS-2). This model has considered an area of 1500m X 1500m with a set of mobile nodes placed randomly and broadcast range is 200m. The simulation was carried out for different number of nodes using Network Simulator (NS2). The performance metrics are packet-delivery ratio, network throughput, delay time and energy level at node. To simulate any networking scenario in network simulator (NS-2) first of all the nodes are created via Tcl script and their initial positions are fixed. The traffic type to be simulated on the network is attached to the node via transport layer agent. On top of this transport layer agent the application layer agents like CBR or FTP are attached.

We have used the IEEE 802.15.4 standard, which specifies the media access control and the physical layer. Val(nn) is the number of nodes, which is set to 30. Val(rp) is set to the NPV protocol, which represent the routing protocol used in the simulation. Val(x) and val(y) are equal to 1500 meter. So 1500 m² is the simulation area. val(stop) represents the simulation time, and is equal to 10 second.

Figure 2 shows output result on Xgraph which plots the packets delivery ratio vs time. In figure 3 graphs shows the performance of NPV protocol in the above generated scenario. X axis represents time in seconds and y axis represents number of packets. The red line shows network throughput and green line represent packet delivery ratio and blue line depicts delay.

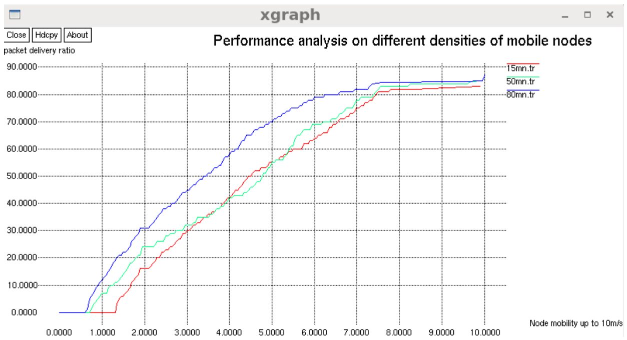


Figure 2: Performance analysis on different density of mobile node without hashing technique

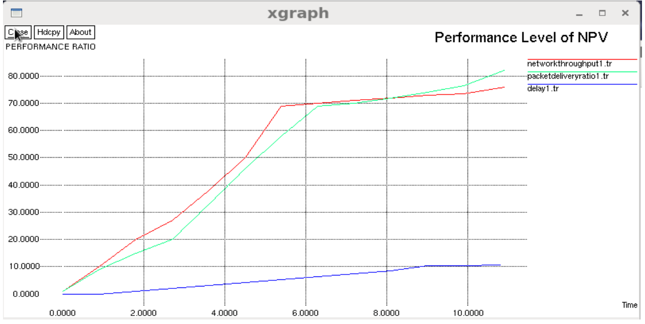


Figure 3: Performance level of NPV without hashing technique

After using enhance message exchange protocol algorithm when we use hashing technique, the Performance is increase on different density of mobile node and in Performance level of NPV, and Because of this the result shows the performance is increase in network throughput, packet delivery ratio and energy level at each node. This difference occurs because, as the traffic flow increases.

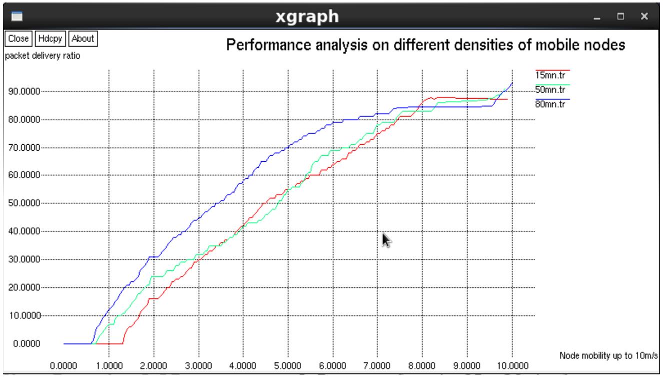


Figure 4: Performance analysis on different density of mobile node with hashing technique

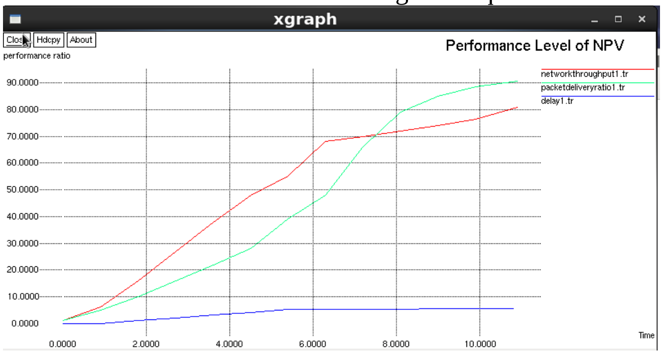


Figure 5: Performance level of NPV with hashing technique

The proposed protocols graph in Figure 5 shows that then the performance of NPV protocol is relatively far better than NPV protocol without using hash technique for transferring the key..

6. Conclusion

This paper predicts an impending crisis in securing Ad Hoc network. Techniques for finding neighbors effectively in a non priori trusted environment are identified. The system eventually provide security as distributed solution as NPV, 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 and analysis showed, and we design algorithm for protocol which is very robust to attacks by independent as well as colluding adversaries, even when they have perfect knowledge of the neighborhood of the verifier. The security solution that achieves both broad protection and desirable network performance is gain. Hence the methods to securing ad hoc network have to be improved. As a conclusion on coordinated attacks, it is the nature of the neighborhood that determines the performance of the NPV scheme in presence of colluders. However, the simulation results in Section 4 show that, in realistic environments, our solution is very robust even to attacks launched by groups of knowledgeable colluders.

References

[1] Marco Fiore,Member, IEEE, Claudio Ettore Casetti, Member, IEEE, Carla-Fabiana Chiasserini,Senior Member, IEEE, and Panagiotis Papadimitratos, Member, IEEE "Discovery and Verification of Neighbor Positions in Mobile Ad Hoc Networks" - Ieee Transactions On Mobile Computing, Vol. 12, No. 2, February 2013.

[2] P. Papadimitratos, M. Poturalski, P. Schaller, P. Lafourcade, D. Basin, S. Capkun, and J.-P. Hubaux, "Secure Neighborhood Discovery:AFundamental Element for Mobile Ad Hoc Networks," IEEE Comm. Magazine, vol. 46, no. 2, pp. 132-139, Feb. 2008.

[3] 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.

[4] 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 , Mar. 2008.

[5] 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.

[6] 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.

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

[8] 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.

[9] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS), 2002.

Author Profile

Pranita D. Pandit, received Bachelor degree of Engineering in Computer Science & Engineering from Sant. GadgeBaba Amravati University in 2010 & Pursuing Master degree of Engineering in Computer Science and Engineering, from P.R.Pote (Patil) College of Engineering and Management, Amravati.

Prof. Pranjali P. Deshmukh, Received Master of Engineering in Computer Science and Engineering from PRMIT&R,Badnera Sant. GadgeBaba Amravati University in 2010. Currently Working as Assistant professor in P.R.Pote (Patil) College of Engineering and Management, Amravati.