ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 9, September 2015

Verifying the Neighbour Positions and Blocking Adversaries in MANET

Niji Gomez, Nishley Elizabeth Joseph

Department of Computer Engineering, Marian Engineering College, University of Kerala, Trivandrum, India

Assistant Professor**,** Department of Computer Engineering, University of Kerala, Trivandrum, India

ABSTRACT: For a continuously self-configuring, infrastructure-less network of mobile devices connected without wires, it is needed to learn the position of their neighbor nodes for the mobile nodes inorder to follow a well organized communication. Specifically, it deals with a Mobile AdhocNETwork, where a pervasive infrastructure is not present, and the location data must be obtained through node-to-node communication. A growing number of ad hoc networking protocols and location-aware services too require that mobile nodes learn the position of their neighbors.However, such a process can be easily distrubted by adversarial nodes. In absence of a priori trusted nodes, the discovery and verification of neighbour positions presents difficulties. 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, it need solutions that let nodes verify the positions of their neighbors, so as to detect adversarial nodes announcing false locations. In this paper, it is presented a solution to discover the position of neighbour nodes securely and verify the discovered positions to know whether the nodes are genuine or not through a Neighbour Position Verification(NPV) protocol.Moreover, it is also proposed to block the adversaries incase of any attacks, thereby restricting their further access to the network.

I. INTRODUCTION

A mobile ad hoc network(MANET) is a collection of wireless mobile nodes that dynamically establishes the network in the absence of fixed infrastructure . It consists of a collection of mobile hosts that may communicate with each another from time to time. No base stations are supported. In Mobile Ad-Hoc Networks, Routes may be disconnected due to dynamic movement of nodes. 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. Each device must forward traffic distinct to its own use, and therefore ,one of the distinctive features of MANET is, each node must be able to act as a router to find out the optimal path to forward a packet. As nodes may be mobile, entering and leaving the network, the topology of the network will change continuously. MANETs provide an emerging technology for civilian and military applications. A MANET is an autonomous collection of mobile users that communicate over relatively bandwidth constrained wireless links. The network is decentralized; where all network activity including discovering the topology and delivering messages must be executed by the nodes themselves.

Since the number of protocols and location aware services used in Mobile Ad Hoc Network (MANET) is increases, the mobile nodes in the MANET wants to find out the position of their neighbors for better and well-organized routing communication. But this process in MANET is easily compromised by attacking mobile node and get information about location of mobile nodes in MANET. Thus we need to have efficient neighbor discover method to prevent such kinds of attacks and give the security. 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, 2) verify the positions of their neighbors, so as to detect adversarial nodes announcing false locations and 3) blocking the neighbour nodes which are faulty in nature to restrict their further access to the network.

Inorder to satisfy these solutions ,the system focus on the latter aspects, hereinafter referred to as neighbor position verification (NPV for short) and blocking adversiaral nodes(BAN for short). NPV procedure enables each node to

DOI: 10.15680/IJIRCCE.2015. 0309142

acquire the locations advertised by its neighbors, and assess their truthfulness. BAN enables to block the neighbour nodes which are faulty in nature and thereby restricting their further access to the network.

II. RELATED WORK

In [8], 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 verification. Furthermore, the resilience of the protocol in [8] to colluding attackers has not been demonstrated. The scheme in [9] 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 [10].

In [11], 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 [11] 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.

The scheme in [12] exploits Time-of-Flight (ToF) distance bounding and node cooperation to mitigate the problems of the previous solutions. However, the cooperation is limited to couples of neighbor nodes, which renders the protocol ineffective against colluding attackers. 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-todevice 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 verification tests to detect independent adversaries, can be found in [13].

III. PROPOSED SYSTEM

A. NPV - An Overview

NPV protocol enables a node, called verifier to discover and verify the position of the neighbor nodes. Inorder to discover the position of the neighbour nodes, a process called message exchange is done between the verifier and the neighbour nodes.Message exchange process contains following messages: poll, reply, reveal and report messages. Verifier finds the transmission time and reception time of messages between the neighbours through the secure message exchange process. Once the message exchange is concluded, the exact distance between the nodescan be calculated easily. Thereafter to verify the position of the corresponding neighbournodes , verification tests like direct symmetry test, cross symmetry test, multi-lateration test are done. Verifier does the verification tests to verify whether the node is verified, faulty or unverifiable.Inorder to restrict the attacks from the faulty nodes,it needs to block the nodes which are faulty in nature. BAN enables to block the neighbour nodes which are faulty in nature and thereby restricting their further access to the network.

B.Message Exchange Process

A source node, S or verifier can initiate the protocol at any time instant, by triggering the 4-step message exchange process [POLL, REPLY,REVEAL and REPORT].

- **POLL message**

A verifier S initiates this message. This message is anonymous. The verifier identity is kept hidden. This carries a public key K’S chosen from a pool of onetime use keys of S.

- **REPLY message**

A communication neighbor Xreceiving the POLL message will broadcast REPLY message after a time interval . This also internally saves the transmission time. It contains some encrypted message with S public key (K’S). This message is called as commitment of X ‘CX’.

- **REVEAL message**

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.

Algorithm 1. Message exchange process: verifier

```
node  S do
        $S \rightarrow *:(POLL,K_S')$
S : store ts
When receive REPLY  from $X \in N_s$ do
 S : store $t_{xs,cx}$
   S :$m_s$={( cx,ix,)|∃txs}
          $S \rightarrow *$ : (REVEAL,ms,Ek’s{ hk’s} ,Sigs,CS)
```

Algorithm 2. Message exchange protocol: any neighbor

```
Forall $X \in NS$ do
when receive  POLL  by S do
  X:storetsx
    X:extract nonce ρx
    X:cx=Ek’s{tsx,ρx}
            $X \rightarrow *$:(REPLY,cx,hk’s)
X:storetx
```

```
when receive REPLY  from Y∈Ns ∩ Nxdo
 X:storetyx,cy
 when receive REVEAL  from S do
    X:ttx={(tyx,iy)|∃tyx}
    X→S:
       (REPORT,EKS{ρx,tx,ttx,px,Sigx,Cx})
```

After this message exchange protocol only the verifier knows all the neighbor nodes position. The verifier notes the transmission time and reception time of poll message, so that it calculates the distance between the nodes. The distance can be calculated by, consider two nodes x and y, the distance between the nodes x and y is,

$$dXY = (tXY - tX) . c$$

where, dXY is the distance between the nodes X and Y, tXY is the actual reception time at y of a message by X, tX is the actual transmission time of a message by X and c is the speed of the message transmission. The distance between y and x is,

$$dYX = (tYX – tY) . c$$

where, dYX is the distance between the nodes Y and X, tYX is the actual reception time at y of a message by Y, tY is the actual transmission time of a message by Y and c is the speed of the message transmission.

C. Position Verification

After calculating the distance between nodes, the verifier does some of the tests to verify whether the node is verified, faulty, or unverifiable. Verified means the node is in current position, faulty means the node is in incorrect position so it may be an adversarial node and unverifiable means the node may be correct or unverifiable. The tests are Direct symmetry test (DST), Cross symmetry test (CST) and Multilateration test (MST). In Direct symmetry test (DST), the distance between the verifier node and other nodes should not exceed twice the ranging error εm, and should be within the error margin $2\varepsilon p + \varepsilon r$ and finally it should not be larger than proximity range R. If the value exceeds, it is noted as faulty. In Cross symmetry test, the verifier node will cross check the information between other two nodes, it will not test the nodes which is ignored in the DST. In CST, pairs of neighbours declaring collinear position with the verifier not taken into account. If there is less than two non-collinear neighbors then the node is unverifiable, if there is more than two non-collinear neighbors, then it is faulty. In Multilaterationtest, the faulty and unverifiable nodes will be avoided, the verifier node links with all the neighbors. It calculates thetransmission time between the verifier node and other neighbours. The points will be noted in the graph and it
gives a hyperbolic curve.each node X for which two or more unnotified links, hencetwo or more hyperbolas exist is considered as suspect.

Algorithm 3. Direct Symmetry Test (DST)

```
node S do
S:Fs←∅
ForallX∈Nsdo
If( |dsx−dxs|>2ϵr+ϵm or
||ps−px || −dsx>2ϵp +ϵr or
Dsx>Rthen
 S: Fs←X
```

Algorithm 4. Cross-Symmetry Test (CST)

```
node S do
$S:Us \leftarrow \emptyset, Ws \leftarrow \emptyset$
Forall $X \in Ns, X \notin Fs$ do
$S: lx=0, mx=0$
 Forall $(X,Y) | X,Y \in Ns, X,Y \notin Fs, X \neq Y$ do
 If $\exists dxy, dyx$ and
 $ps \notin line(px,py)$ then
$S:lx=lx+1, ly=ly+1$
 If( $|dxy - dyx| > 2\epsilon r + \epsilon m$ or
 $px - py - dxy > 2\epsilon p + \epsilon r$ or
 $dxy > R$ )then
  $S:mx=mx+1, my=my+1$
  Forall $X \in Ns, X \notin Fs$ do
  $if$ $lx<2$ then $S:Us \leftarrow X$
 $elseswitch$ $mx/lX$ do
$case$ 1: $mx/lX > \delta$   $S:FS \leftarrow X$
$case$ 2: $mx/lX = \delta$   $S:US \leftarrow X$
$case$ 3: $mx/lX < \delta$   $S:WS \leftarrow X$
```

Algorithm 5. Multilateration Test (MLT)

```
node S do
 $S:Vs \leftarrow \emptyset$
 Forall $X \in Ws$ do
 $S:Lx \leftarrow \emptyset$
 Forall $(X,Y) | X,Y \in Ws, X \neq Y$ do
 If $\exists txy$ and $\nexists tyx$ then
 $S:Lx \leftarrow Lx(S,Y)$
 Forall $X \in Ws$ do
 $if$ $|Lx|>2$ then
 $S: pX^{ML} = argmin np \sum Li, \in LX \| p - Li \cap Lj \|^2$
$if$ $\| pX - pX^{ML} \| > 2\epsilon p$ then
$S:FS \leftarrow X, = WS \setminus X$
 $S:VS=WS$
```

D. BAN Procedure

In this the verifier blocks the neighbour nodes which shown as faulty in the verification tests. This is done to avoid the further access to the network by the adversiaral nodes in future. This can reduce the attack by the independent as well as the colluding adversaries in future.

The BAN procedure can be done by sending a blocking command to the server with target client ip. The server then blocks the client. And for the future reference , to know whether a node is already blocked or not ,a list of blocked ip's are stored in a file on server. When the client application starts , the server check for blocked or not. If it is blocked then exits from the application.

E.Chat Application

Once the verifier tagged a neighbour node as verified , the corresponding node can send messages to everyone in the network. And the neighbour node who is faulty in nature or doing any malpractice is get blocked by the server and exits from the application. There is also a possibility of viewing the conversation between the verified nodes in future if it is saved by the corresponding nodes.

IV. PERFORMANCE ANALYSIS

Here thesecurity comes at a cost as the adversaries gets blocked by the verifier in the initial run itself. This could minimize the time taken to test a neighbour at each and every minute they enter the network.Here once a neighbour been tagged as faulty, it gets blocked by the verifier and restricting their further access to the system.This could improve the performance of the system as it reduced the time taken for repeated tests of the same neighbour at every approach of entering into the network. In the existing systems, there is no such technique to block the adversaries in case of any attack, thereby restricting their further access to the network. Hence ,it is needed to do the verification step whenever the neighbour node enter into the network.In the graph it is plotted the response time taken by the verifier inorder to verify the neighbour node in the existing systems (nonsecure protocol) as well as the proposed system(secure protocol).

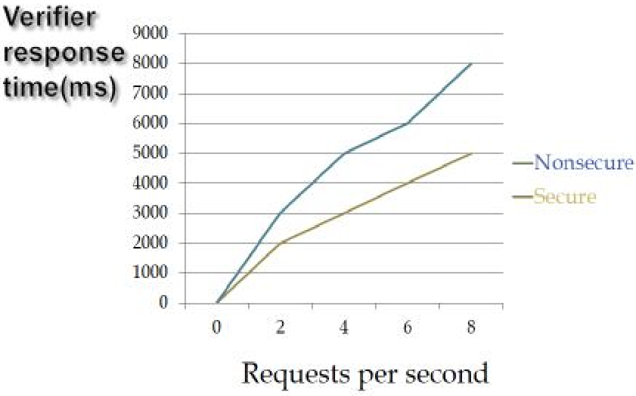

Figure 7.1. Comparison graph between secure and nonsecure systems

V. CONCLUSION

The reportcompared various techniques related to Neighbour Position Verification and presented a distributed solution for NPV, which allows any node in a mobile ad hoc network to verify the positionof its communication neighbors without relying on a priori trustworthy nodes. And also a solution to block the neighbour nodes which found faulty in verification process, thereby restricting their further access to the network. The analysis showed that NPV protocolis very robust to attacks by independent as well as colluding adversaries, even when they have perfect knowledge of theneighborhood of the verifier. The solution is also effective in identifying nodes advertisingfalse positions, while keeping the probability of false positives low. Simulation results confirm that the system takes only few milliseconds to verify the neighbour node position as it is need not to test the neighbour nodes which are already tagged as faulty.

REFERENCES

[1]. Marco Fiore, *Member, IEEE*, Claudio EttoreCasetti, *Member, IEEE*, Carla-FabianaChiasserini, *Senior Member, IEEE*, and PanagiotisPapadimitratos, *Member, IEEE*," Discovery and Verification of Neighbor Positions in Mobile Ad Hoc Networks",*IEEE* 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] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Packet Leashes: A Defense against Wormhole Attacks in Wireless Networks," *Proc. IEEE INFOCOM*, Apr. 2003.

[4] J. Eriksson, S. Krishnamurthy, and M. Faloutsos, "TrueLink: A Practical Countermeasure to the Wormhole Attack in Wireless Networks," *Proc. IEEE 14th Int'l* Conf. Network Protocols (ICNP), Nov. 2006.

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

[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] S. _Capkun, 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.

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

[9] 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.

[10] 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 AdHoc Networks (VANET)*, Sept. 2006.

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