International Journal of Emerging Trend in Engineering and Basic Sciences (IJEEBS)
ISSN (Online) 2349-6967 *Volume 2 , Issue 1(Jan-Feb 2015), PP1-5*

Discovery and Verification of Neighbor Position in Ad Hoc Network

Nikhil Jadhav[1], Kushal Bang[2], Anurag Boggavarapu[3], Akshay Gaikwad[4], Zarina Y. Shaikh[5]
[1.2.3.4.5] *(Computer, Rajarshi Shahu College of Engineering.)*

Abstract :- Discovery of position has become important in today's world due to the growing technologies in mobile system. However, due to presence of adversarial nodes, it has become difficult to find out trusted nodes with true position. In this paper we are dealing with this problem and finding secure nodes with true positions. We will be using neighbor position verification protocol (NPV) for solving this problem.

Keywords: - Neighbor position verification, ad hoc network.

I. INTRODUCTION

In fast growing technologies there are many applications which require location awareness that has become a need in mobile system. In robotics to coordinate the movement of the nodes, in geographic routing in spontaneous networks and traffic monitoring in vehicular networks are some of the fields which require location awareness of all the neighboring nodes.

As the nodes consists of mobile which are dynamic in nature. The network topology is subjected to change rapidly and unpredictably over time. As it is an unstructured network. It doesn't work continuously under any topology. The network is decentralized where all network activity including discovering, topology and delivering message must be executed by the nodes themselves. Neighbor discovery (ND) provides an important functionality for wireless devices that is to discover other devices that they can communicate directly through the wireless networking. Routing begin the most important in the context of wireless communication makes it easy to abuse ND. The verification of node locations is an important issues in mobile networks, and it becomes particularly challenging in the presence of adversaries aiming at harming the system.

Therefore we need a solution to 1) to correctly establish their location in spite of attack feeding false location information, 2) and verify the position of neighbor so as to detect adversarial nodes.

Here in this paper, we will be using neighbor position verification (NPV) protocol for node discovery and verification in ad hoc network. The NPV is compatible with the state-of-the-art security architecture, which also include the one that has been proposed for vehicular networks [1], [2].

II. SYSTEM AND ADVERSARIAL MODEL

We consider a mobile network and consider its node as a communicating neighbor if it can reach other nodes directly [2]. Here we are assuming that every node knows its position in the network with some maximum error e_P and has a common time reference with other nodes in the network. This can be achieved by equipping the nodes with GPS receiver. Further we assume that the nodes can perform TOF based RF ranging with some maximum error e_r. The techniques for calculating TOF based RF ranging is described in [3].

Let e_m be the tolerance value for the nodes with relative fast movement. Because the node do not move significantly during protocol execution, and during the execution the message exchange do not take more than few milliseconds.

We assume each node owns a set of private key, kx, and public key, Kx, and set of one time use key {kx', Kx'} as mentioned in emerging architecture for secure and privacy-enhancing communication [2], [4]. Nodes X can encrypt and decrypt data with its key and public key of other nodes and can produce digital signature (Sigx)

with its private key. Nodes can authenticate messages of others nodes through public key cryptography as mentioned in [5].

Nodes are assumed to be true if they act in accordance with NPV protocol otherwise adversarial if they do not.

III. NPV: AN OVERVIEW

NPV is a protocol which consists of a node, hereinafter called as verifier which is used to verify and discovery the position of its communication nodes. The verifier is used for starting the NPV protocol. Protocol uses set of 4 messages for discovering the position of the communicating nodes. The purpose of the messages is for getting the information about the two communicating nodes which we can use for finding out distance between them. The 4 messages used are poll, reply, reveal and report which are described in next section. After collecting the information verifier uses timing to perform TOF-based ranging and computes distance between all pairs of nodes in network [3]. After calculating the distance we classify the nodes into 3 parts for completing the verification process. The 3 parts are verified nodes, i.e., which are safe for communication, faulty nodes, i.e., which are not safe for communication, then the unverified node, i.e., which cannot be proved to be safe or not. The process of verification is carried out with help of two tests the direct symmetry test and cross symmetry test. Thus after carrying out the process correctly we can easily avoid the adversary from entering the network for communication and secure the network from such adversary.

IV. NPV PROTOCOL

MESSAGE EXCHANGE

We consider following notations for message exchange protocol.
Px = current position of node X.
Tx = time at which node X sends message.
Txy = time at which node Y receives message send by node X.
To avoid latency in actual time calculation we use the implementation specified in [6]. We use the following algorithms where algorithm 1 is used by the verifier node, i.e., S and algorithm 2 is used by the communicating neighbors of S.

Algorithm 1: Message Exchange Protocol (MEP): verifier

1. If (node S) do
S →∗: (POLL, KS′)
S: store tS)
2. If (REPLY received fromX ∈ NS) do
S: store tXS, tX,
end if
3. if(Tmax+ Δ + Tjitter do
S: mS= {(tX, iX,)|∃ tXS}
S →∗: (REVEAL, mS, EKS′ hKS′ , Sigs, CS)
end if

Algorithm 2: MEP for Any neighbor
1. ForX ∈ NS do
2. If (POLL receive by S) do
X: store tXS
X: extract TX uniform r. v. ∈ [0, Tmax]

End if
3. After TX do
X: extract nonce ρx
X: tX = EKS′ {tXS , ρx}
X →∗: (REPLY , tX , hKS′)
X: store tXS
4. If (REPLY received from Y ∈ NS ∩ NX) do
X: store tYX , tY
End if
5. If (REVEAL received from S) do
X: tX = {(tYX , iY)|∃tYX}
X → S: (REPORT, EKS {ρx, tX , ttX , px, SigX , CX })
End if
6. End for

4.1.1 POLL MESSAGE

The NPV protocol is initiated by broadcasting the POLL message to all communicating neighbors of S. The POLL message does not carry the identity of the verifier it uses freshly generated Mac address. It uses public key K's from pool of one time used key of node S.

4.1.2 REPLY MESSAGE

The other communicating neighbors receive the POLL message and save the time at which they receive the message. After that these communicating neighbor's broadcasts REPLY message which consist of freshly generated Mac address and some information encrypted with the public key (K's) received from the POLL message.

4.1.3 REVEAL MESSAGE

REVEAL message is broadcasted by the verifier S to all the communicating nodes with the real MAC address of the verifier S. 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.

4.1.4 REPORT MESSAGE

All the communicating neighbors X send the REPORT message to the verifier. This message contains the position of node X and the transmission time of node X's REPLY message, and all node's temporary identifiers and reception time received in REPLY message of X's neighbor. The identifier is obtained from the MAP in the REVEAL message. And the node X discloses its identity by sending its digital signature and certified public key.

4.2 POSITION VERIFICATION

Once the message exchange has taken place. The verifier decrypt the data received and find the position of all the nodes participating in the network. Using the ToF-based technique, S finds out the distance from all communicating neighbor in the network. The distance calculation is done using this expression: dXY = (tXY-tX) * c, where c is the speed of light and (X, Y) is pair of communicating neighbor such that X, Y ∈ NS and the timing information is obtained from previous message exchange between nodes.
Then the distance computed is processed through following two verification tests
1. Direct Symmetry Test (DST)
2. Cross Symmetry Test (CST)

Algorithm 3: DST
1. If node S do
S: Fs ← ∅
2. For X ∈ NS do
3. If(|dSX − dXS| > 2ϵr + ϵm or || ps − px| − dSX| > 2ϵp + ϵrordSX > R)$then$
S: Fs ← ∅

DST is the first verification test which is performed. In DST, verifier checks the direct links with its communicating neighbor's. In first step it verifies that the distances dSX and dXS, obtained from ranging do not differ by more than twice the ranging error plus a tolerance value. In second step it verifies that the position advertised by the neighbor is consistent with such distances, within an error margin. Lastly the verifier checks if the distance is not larger than the range of network. The verifier enters the node into set of faulty nodes if a mismatch is found with any of these checks.

Algorithm 4: CST
1. If node S do
S: Us ← ∅, Ws ← ∅
2. For X ∈ NS , X ∉ FS do
S: lX = 0, mX = 0
End for
3. For (X, Y)|X , Y ∈ NS , X, Y ∉ FS , X ≠ Y do
4. If ∃ dXY − dYX and ps ∉ line px, py then
S: lX = lX + 1, lY = lY + 1
5. If(|dXY − dYX | > 2ϵr + ϵm or ||pX − pY| − dXY| > 2ϵp + ϵr or dXY > R)$then$
S: mX = mX + 1, mY = mY + 1
End if
End if
End for
6. For X ∈ NS , X ∉ FS do
7. If lX < 2 $then$
S: US ← X
8. $else\ switch$ (mx/) do
$case$ 1: (mx/lX) > δ
S: FS ← X
$case$ 2: (mx/lX) = δ
S: US ← X
$case$ 3: (mx/lX) < δ
S: WS ← X
End if-else
End for
End if

CST is the second test which is carried out. It checks on the information mutually gathered by each pair of communication neighbor's. The CST ignores nodes which are declared as faulty by DST and only considers nodes that provided to be communication neighbor between each other. However, pairs of neighbor declaring collinear positions with respect to verifier are not considered. For all other nodes CST carries the checks as in DST with that it also maintain a link counter and a mismatch counter. The link counter is incremented on every new cross check and mismatch counter is incremented if at least one of the checks on distance and position fails. If a node shares less than two non-collinear neighbor's with verifier it is tagged as unverified node. While the

other nodes are classified on bases of the ratio of mismatch counter to link counter. Ratio is compared with a constant value whether it is greater than or equal to or less than the constant value. If greater than constant value then tagged as faulty nodes, else if less than constant value then tagged as conditionally verified node.

V. CONCLUSION

Techniques for finding neighbor's effectively in a non priori trusted environment are identified. The proposed techniques will eventually provide security from malicious nodes. The protocol is robust to adversarial attacks. This protocol will also update the position of the nodes in an active environment. The performance of the proposed scheme will be effective one. This proposed protocol is capable of positioning of nodes which are not linearly position, but are deviated from one another. Hence in future scope we will try to propose a solution to this problem.

REFERENCES

[1] 1609.2-2006: IEEE Trial-Use Standard for Wireless Access in Vehicular Environment – *Security Services for Applications and Manager Messages*, IEEE, 2006

[2] P.Papadimitratos, M.Poturalski, P.Lafourcade, D.Basin, S.Capkum, and J-P, Hubaux, *Secure Neighborhood Discovery: A Fundamental Element for Mobile Ad-Hod Networks*," IEEE comm.Magazine, *vol.46, no.2, pp.132-139*, Feb.2008 (neighbor discovery based on distance). (9)

[3] S.Capkun and J-P.Hubaux, "*Secure Positioning in Wireless Networks*," IEEE J.Selected Areas in Comm., *Vol.24, pp.221-232*, Feb.2009.

[4] PRECIOSA: Privacy Enabled Capability in Co-Operative Systems and Safety Applications, http://www.preciosa-project.org, 2012.

[5] G. Calandriello, P. Papadimitratos, A. Lioy, and J.-P. Hubaux, "*On the Performance of Secure Vehicular Communication Systems*," IEEE Trans. Dependable and Secure Computing, *vol. 8, no. 6, pp. 898-912*, Nov/Dec. 2011.

[6] M. Ciurana, F. Barcelo-Arroyo, and F. Izquierdo, "*A Ranging System with IEEE 802.11 Data Frames*," Proc. IEEE Radio and Wireless Symp. Jan. 2007.