Journal of Mobile Communication 1 (2): 75-84, 2007


# Secured on Demand Position Based Private Routing Protocol for Mobile AdHoc Network-SO2P

[1]M. Jaiganesh and [2]D. Gunaseelan
[1]P.S.N.A College of Engineering and Technology, Dindigul, Tamilnadu, India
[2]Gandhigram University, Dindigul, Tamilnadu, India

**Abstract:** Privacy is needed in adhoc networks. A secured on demand position based private routing algorithm is proposed for communication anonymity and provide a security in mobile Adhoc network. Here we use cryptography algorithm for providing security to prevent message hacking information from internal and external attackers. The position information of nodes is drawn by GPS (Global Positioning System). Here, we provide a terminal node behave like server to getting position information of nodes in the network. A Simulator based on GLOMOSIM is developed for evaluating security. It provides a secure data transmission between the nodes. In our design, the server with the process of GPS system records all the information about the terminal nodes involved in the Ad-hoc networks. This server will send the Position Reply (PREP) to the source when it receives the Position Request (PREQ). This PREP is encrypted to provide the security and it's transmitted to all the intermediate nodes in the Ad-hoc networks and the intermediate performs decryption and gets the destination address. The intermediate nodes checks with its own address and if it matches it request for the data and it sends Node Reply, else it will again sends the PREP to all the other intermediate nodes. This Node Reply is then analyzed to find the optimal route. In this paper we proposed a new protocol using Cryptographic techniques. GLOMOSIM-(A Global Mobile Simulator) is developed for evaluating the security.

**Key words:** Ad hoc routing protocol, security, privacy, anonomity, cryptographic algorithm

## INTRODUCTION

An ad hoc network is often defined as an infrastructureless network, meaning a network without the usual routing infrastructure like fixed routers and routing backbones. Typically, the ad hoc nodes are mobile and the underlying communication medium is wireless. Each ad hoc node maybe capable of acting as a router. Such ad hoc networks may arise in personal area networking, meeting rooms and conferences, disaster relief and rescue operations, battlefield operations, etc. Some aspects of ad hoc networks have interesting security problems (Ramanathan and Steenstrup, 1996; Royer and Toh, 1999; Asokan and Ginzboorg, 2000). Routing is one such aspect. Several routing protocols for ad hoc networks have been developed, Particularly in the MANET working group of the Internet Engineering Task Force (IETF).

Surveys of routing protocols for ad hoc wireless networks are presented in this study, we consider the security of routing protocols for ad hoc networks.

## RELATED WORK

There is very little published prior work on the security issues in ad hoc network routing protocols. Neither the survey by Ramanathan and Steenstrup (1996) nor the survey by Royer and Toh (1999) mention security. None of the draft proposals in the IETF MANET working group have a non-trivial security considerations section. Actually, most of them assume that all the nodes in the network are friendly and a few declare the problem out-of-scope by assuming some canned solution like IPSec may be applicable. There are some works on securing routing protocols for fixed networks that also deserved to be mentioned here.

Perlman (1983) in her thesis, proposed a link state routing protocol that achieves Byzantine Robustness. Although her protocol is highly robust, it requires a very high overhead associated with public key encryption. Secure BGP (Kent *et al*., 2000) attempts to secure the Border Gateway Protocol by using PKI (Public Key Infrastructure) and IPsec. In their paper on securing ad hoc networks (Zhou and Haas, 1999).

Zhou and Haas primarily discuss key management and they devote a section to secure routing, but essentially conclude that nodes can protect routing information in the same way they protect data traffic. They also observe that denial-of-service attacks against routing will be treated as damage and routed around.

**Corresponding Author:** M. Jaiganesh, P.S.N.A College of Engineering and Technology, Dindigul, Tamilnadu, India

Security issues with routing in general have been addressed by several researchers (Smith *et al*., 1997; Hauser *et al*., 1997). And, lately, some work has been done to secure ad hoc networks by using misbehavior detection schemes (Marti *et al*., 2000). This approach has two main problems: first, it is quite likely that it will be not feasible to detect several kinds of misbehaving (especially because it is very hard to distinguish misbehaving from transmission failures and other kind of failures) and second, ithas no real means to guarantee the integrity and authentication of the routing messages.

Dahill *et al*. (2001) proposed ARAN, a routing protocol for ad hoc networks that uses authentication and requires the use of a trusted certificate server. In ARAN, every node that forwards a route discovery or a route reply message must also sign it, (which is very computing power consuming and causes the size of the routing messages to increase at each hop), whereas the proposal presented in this study only require originators to sign the message. In addition, it is prone to reply attacks using error messages unless the nodes have time synchronization. Papadimitratos and Haas (2002) proposed a protocol (SRP) that can be applied to several existing routing protocols (in particular DSR (Johnson *et al*., 2002) and IERP (Hass *et al*., 2002). SRP requires that, for every route discovery, source and destination must have a security association between them. Furthermore, the study does not even mention route error messages. Therefore, they are not protected and any malicious node can just forge error messages with other nodes as source. Hash chains have being used as an efficient way to obtain authentication in several approaches that tried to secure routing protocols. In (Cheung, 1997; Hauser *et al*., 1997; Perrig *et al*., 2001) they use them in order to provide delayed key disclosure. While, in Zhang (2001), hash chains are used to create one-time signatures that can be verified immediately. The main drawback of all the above approaches is that all of them require clock synchronization.

In SEAD (2002) (by Hu *et al*., 2001) hash chains are also used in combination with DSDV-SQ (Broch *et al*., 1998) (this time to authenticate hop counts and sequence numbers). At every given time each node has its own has chain. The hash chain is divided into segments; elements in a segment are used to secure hop counts in a similar way as we do in SAODV. The size of the hash chain is determined when it is generated. After using all the elements of the hash chain a new one must be computed. SEAD can be used with any suitable authentication and key distribution scheme. But finding such a scheme is not straightforward. In the present study we suggest some non-standard approaches that can be used to achieve key distribution. Ariadne (Hue *et al*., 2001) by the same authors, is based on DSR (Johnson *et al*., 2002) and TESLA (Perrig *et al*., 2001) (on which it is based its authentication mechanism). It also requires clock synchronization, which we consider to be an unrealistic requirement for ad hoc networks. It is quite likely that, for a small team of nodes that trust each other and that want to create an ad hoc network where the messages are only routed by members of the team, the simplest way to keep secret their communications is to encrypt all messages (routing and data) with a team key. Every member of the team would know the key and, therefore, it would be able to encrypt and decrypt every single packet. Nevertheless, this does not scale well and the members of the team have to trust each other. So it can be only used for a very small subset of the possible scenarios. Looking at the work that had been done in this area previously, we felt that the security needs for ad hoc networks had not been yet satisfied (at least for those scenarios where everybody can freely participate in the network). In the next section, we specify that what are those needs in the format of a list of security requirements

## ATTACKS ON AD HOC NETWORKS

Attacks on ad hoc network routing protocols generally falls into one of two categories:

**Routing-disruption attacks:** The attacker attempts to cause legitimate data packets to be routed in dysfunctional ways.

**Resource-consumption attacks:** The attacker injects packets into the network in an attempt to consume valuable network resources such as bandwidth or to consume node resources such as memory (storage) or computation power. From an application-layer perspective, both attacks are instances of a Denial-of-Service (DoS) attack. An example of a routing-disruption attack is for an attacker to send forged routing packets to create a routing loop, causing packets to traverse nodes in a cycle without reaching their destinations, thus consuming energy and available bandwidth. An attacker might similarly create a routing black hole, which attracts and drops data packets.

An attacker creates a black hole by distributing forged routing information (that is, claims ng falsified short distance information); the attacker attracts traffic and can then discard it. In a special case of a black hole, an attacker could create a gray hole, in which it selectively drops some packets but not others, for example, by forwarding routing packets but not data packets. An attacker also might attempt to cause a node to use a route

detour (suboptimal routes), or partition the network by injecting forged routing information to prevent one set of nodes from reaching another. An attacker might attempt to make a route through itself appear longer by adding virtual nodes to the route; we call this attack a *gratuitous detour* because a shorter route exists and would otherwise have been used. In ad hoc network routing protocols that attempt to keep track of perceived malicious nodes in a blacklist at each node, such as is done in the watchdog and path rater protocol, an attacker might malign a good node, causing other good nodes to add that node to their blacklists and thus avoid that node in future routes. A more subtle type of routing-disruption attack is creating a wormhole in the network, using a pair of attacker nodes A and B linked via a private network connection. We give an example of this attack and a countermeasure in the next study.

A rushing attack is a malicious attack that is targeted against on-demand routing protocols that use duplicate suppression at each node2 an attacker disseminates route requests quickly throughout the network, suppressing any later legitimate route requests when nodes drop them due to the duplicate suppression

## DESIGN OF THE S02P

Our project design involves the designing of the following modules:

- Position tracking module.
- Secured key generating module.
- Broadcast route initiation module.
- Broadcast route reply module.
- Optimal path attack preventing module.

**Position tracking module:** In this first module, the source is supposed to get the destination information from the server. The server will send the desired destination address with its position to the requested source. The server will record all the information about its terminal nodes that are involved in its Ad-hoc networks and that information includes terminal address and its position. The server with the process of GPS system does this information tracking. Here the source requests the position of the destination, which is Position Request (PREQ) to the server. The server with its recorded information sends the reply to the source, which is Position Reply (PREP) to the source, which includes Destination address with its position. This position reply will be the input to the second secured key generating module, which generates a secured key for the data transmission to the desired destination (Fig. 1).

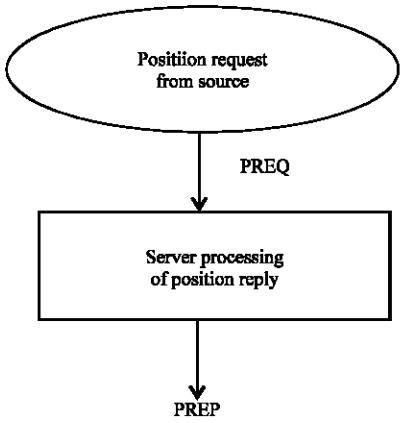


Fig. 1: Position tracking

**Data Encryption Standard (DES):** In the next module, namely Secured key generating module, I am going to make secured key by using DES algorithm. A DES key consists of 64 binary digits ("0"s or "1"s) of which 56 bits are randomly generated and used directly by the algorithm. The other 8 bits, which are not used by the algorithm, may be used for error detection. The 8 error detecting bits are set to make the parity of each 8-bit byte of the key odd, i.e., there is an odd number of "1"s in each 8-bit byte1. A TDEA key consists of three DES keys, which are also referred to as a key bundle. Authorized users of encrypted computer data must have the key that was used to encipher the data in order to decrypt it. Security of the data depends on the security provided for the key used to encipher and decipher the data. Data can be recovered from cipher only by using exactly the same key used to encipher it. Unauthorized recipients of the cipher who know the algorithm but do not have the correct key cannot derive the original data algorithmically. However, it may be feasible to determine the key by a brute force exhaustion attack. Also, anyone who does have the key and the algorithm can easily decipher the cipher and obtain the original data. A standard algorithm based on a secure key thus provides a basis for exchanging encrypted computer data by issuing the key used to encipher it to those authorized to have the data. Data that is considered sensitive by the responsible authority, data that has a high value, or data that represents a high value should be cryptographically protected if it is vulnerable to unauthorized disclosure or undetected modification during transmission or while in storage. This algorithm is designed to encipher and decipher blocks of data consisting of 64 bits under control of a 64-bit key1.

Deciphering must be accomplished by using the same key as for enciphering, but with the schedule of

addressing the key bits altered so that the deciphering process is the reverse of the enciphering process. A block to be enciphered is subjected to an initial permutation IP, then to a complex key-dependent computation and finally to a permutation which is the inverse of the initial permutation IP-1. The key-dependent computation can be simply defined in terms of a function f, called the cipher function and a function KS, called the key schedule.

## DES ENCRYPTION

The 64 bits of the input block to be enciphered are first subjected to the following permutation, called the initial permutation IP (Fig. 2):

IP
58 50 42 34 26 18 10 2
60 52 44 36 28 20 12 4
62 54 46 38 30 22 14 6
64 56 48 40 32 24 16 8
57 49 41 33 25 17 9 1
59 51 43 35 27 19 11 3
61 53 45 37 29 21 13 5
63 55 47 39 31 23 15 7

That is the permuted input has bit 58 of the input as its first bit, bit 50 as its second bit and so on with bit 7 as its last bit. The permuted input block is then the input to a complex key-dependent computation described below. The output of that computation, called the preoutput, is then subjected to the following permutation which is the inverse of the initial permutation:

IP-1
40 8 48 16 56 24 64 32
39 7 47 15 55 23 63 31
38 6 46 14 54 22 62 30
37 5 45 13 53 21 61 29
36 4 44 12 52 20 60 28
35 3 43 11 51 19 59 27
34 2 42 10 50 18 58 26
33 1 41 9 49 17 57 25

That is, the output of the algorithm has bit 40 of the preoutput block as its first bit, bit 8 as its second bit and so on, until bit 25 of the preoutput block is the last bit of the output. 11 The computation which uses the permuted input block as its input to produce the preoutput block consists, but for a final interchange of blocks, of 16 iterations of a calculation that is described below in terms of the cipher function f which operates on two blocks, one of 32 bits and one of 48 bits and produces a block of 32 bits. Let the 64 bits of the input block to an iteration consist of a 32 bit block L followed by a 32 bit block R.

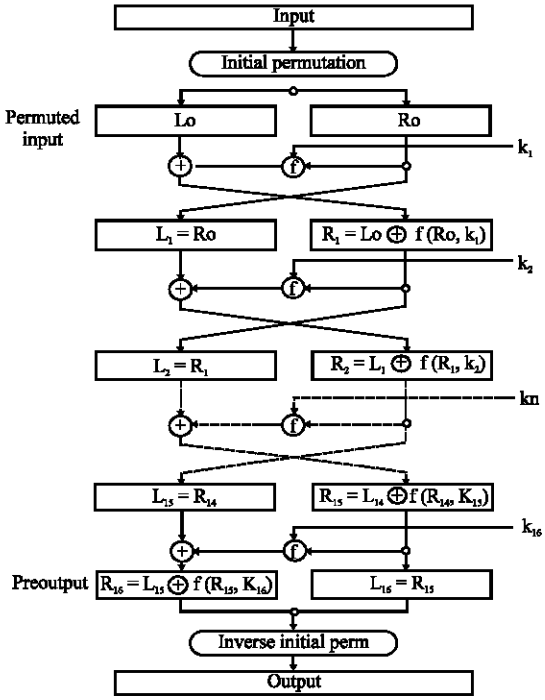


Fig. 2: DES computation

Using the notation defined in the introduction, the input block is then LR. Let K be a block of 48 bits chosen from the 64-bit key. Then the output L'R' of iteration with input LR is defined by:

L' = R
R' = L Å f(R, K)

Where Å denotes bit-by-bit addition modulo 2. As remarked before, the input of the first iteration of the calculation is the permuted input block. If L'R' is the output of the 16th iteration then R'L' is the preoutput block. At each iteration a different block K of key bits is chosen from the 64-bit key designated by key. With more notations we can describe the iterations of the computation in more detail. Let KS be a function which takes an integer n in the range from 1 to 16 and a 64-bit block KEY as input and yields as output a 48-bit block Kn which is a permuted selection of bits from key. That is

Kn = KS (n, KEY)

With Kn determined by the bits in 48 distinct bit positions of KEY. KS is called the key schedule because the block K used in the n'th iteration of (1) is the block Kn determined by (2). As before, let the permuted input block be LR. Finally, let L () and R() be, respectively L and R and let Ln and Rn be, respectively L' and R' of (1) when L and R are, respectively Ln-1 and Rn-1 and K is Kn; that is, when n is in the range from 1 to 16.

3. Ln = Rn-1
Rn = Ln-1 Å f (Rn-1, Kn)

The preoutput block is then R16L16. The key schedule KS of the algorithm is described in detail in the Appendix. The key schedule produces the 16 Kn which are required for the algorithm.

## DES DECRYPTION

The permutation IP-1 applied to the preoutput block is the inverse of the initial permutation IP applied to the input. Further, from (1) it follows that:

4. R = L'
L = R' Å f (L', K)

Consequently, to decipher it is only necessary to apply the very same algorithm to an enciphered message block, taking care that at each iteration of the computation the same block of key bits K is used during decipherment as was used during the encipherment of the block. Using the notation of the previous section, this can be expressed by the equations:

5. Rn-1 = Ln
Ln-1 = Rn Å f (Ln, Kn)

Where now R16L16 is the permuted input block for the deciphering calculation and L0R0 is the preoutput block. That is, for the decipherment calculation with R16L16 as the permuted input, K16 is used in the first iteration, K15 in the second and so on, with K1 used in the 16th iteration.

**The cipher function f:** A sketch of the calculation of f(R, K) is given below, Let E denote a function which takes a block of 32 bits as input and yields a block of 48 bits as output (Fig. 3). Let E be such that the 48 bits of its output, written as 8 blocks of 6 bits each, are obtained by selecting the bits in its inputs as follows:

**E bit-selection table:**

32 1 2 3 4 5
4 5 6 7 8 9
8 9 10 11 12 13
12 13 14 15 16 17
16 17 18 19 20 21
20 21 22 23 24 25
24 25 26 27 28 29
28 29 30 31 32 1
14

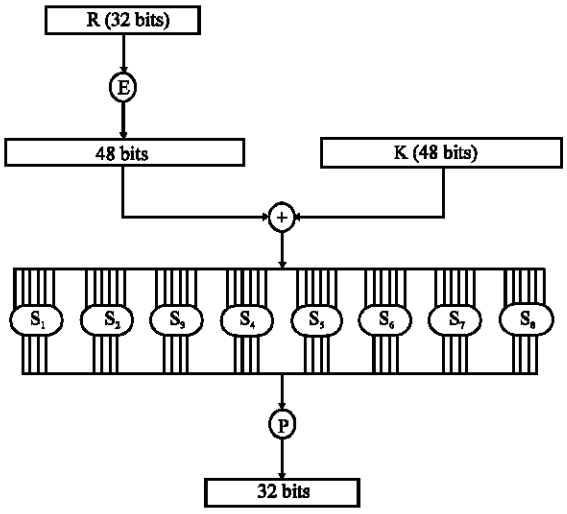


Fig. 3: Key schedule calculation

Thus the first 3 bits of E(R) are the bits in positions 32, 1 and 2 of R while the last 2 bits of E(R) are the bits in positions 32 and 1. Each of the unique selection functions S1, S2...S8 takes a 6-bit block as input and yields a 4-bit block as output and is illustrated by using a table containing the recommended S1:

**Secured key generating module:** In this module, the input is the position Reply from the first module, which includes the destination address and its position. This module done the process of secured key generation with the destination address and the position, which achieves the security for the data transmission to the destination. The DES algorithm does the security, which is one of the best Ad-hoc security algorithms for protecting against the internal attackers from hacking the information about the destination and its data. The position reply from the first module is given to the secured key-generating mode, which generates the key by using the destination secured encrypted cipher text is broadcasted to all the intermediate terminal nodes involved in the ad-hoc networks and this process is said to be Broadcast Route initiation process. The output of the module is the broadcasting of the Secured cipher text to all intermediate nodes (Fig. 4 and 5).

**Broadcast route initiation module:** In this module, the cipher text from the source is the input to all the intermediate nodes involved in the ad-hoc networks. This cipher text is the input to the secured key decryption DES mode. This cipher text is decrypted with the Same DES security algorithm, which finds the plain text. This plain text is the desired destination address for which the source needs to send the desired data.

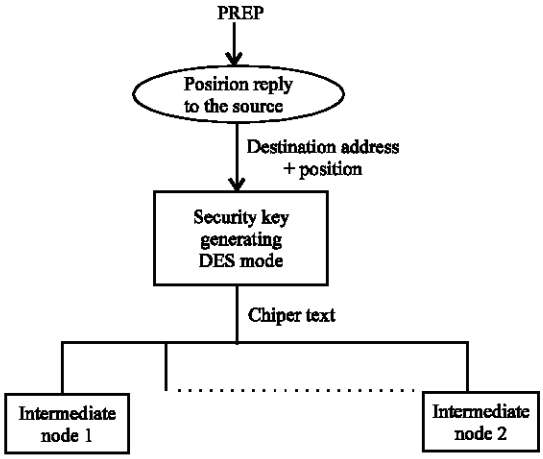


Fig. 4: Secured key generating

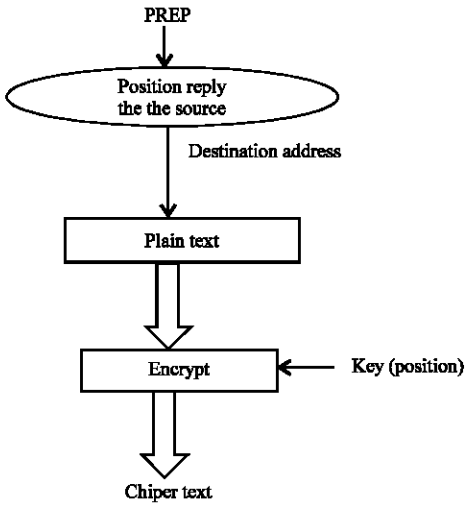


Fig. 5: Secured key generating DES mode

Now this plain text is the output of this module, which is the input to the next Broadcast route Reply module (Fig. 6).

**Broadcast route reply module:** In this module, the plain text from the third module, which is done by secured Decryption DES mode. Now in this module, the plain text is used to check whether the plain text is equal to the desired destination and this is done by the destination address checking mode. The destination address checking modes get the plain text as its input. This mode checks whether the plain text is equal to the destination address and reply that information to the requested source. If that plain text is equal to the requested destination address, then the intermediate node will request the data to the source, by sending the route reply

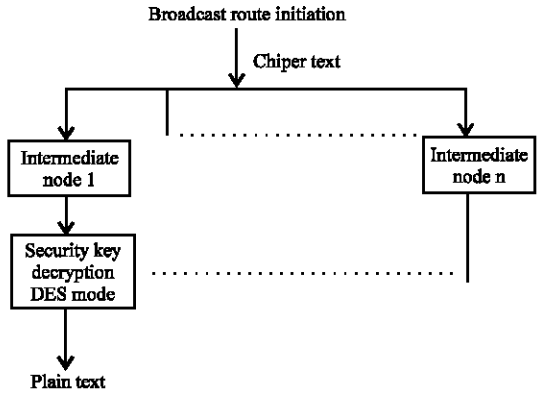


Fig. 6: Broadcast route initiation

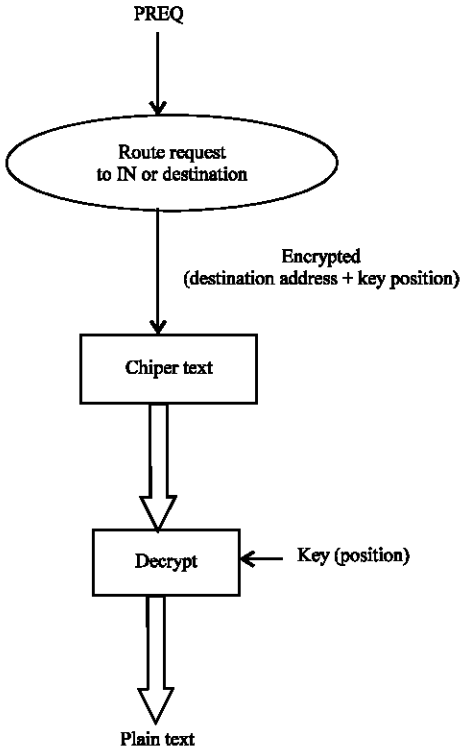


Fig. 7: Secured key decryption DES mode

with its matched destination address. So, now this node reply to the source is the output of this module, which is given as the input to the next optimal path attack-preventing module (Fig. 7-9).

**Optimal path attack preventing module:** In this module, the node reply is the input to the source, which is received from the intermediate node. This node reply is given to the optimal route analysis mode. This optimal route analysis mode does the process of selecting the best optimal path for sending the desired data to the

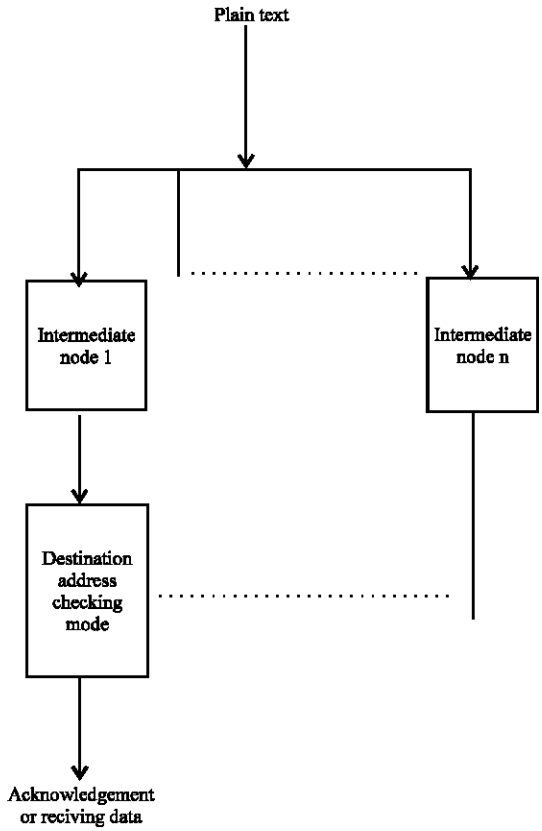


Fig. 8: Broadcast route reply

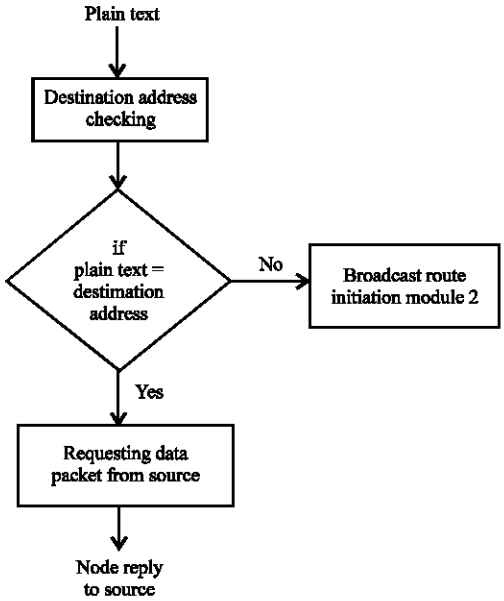


Fig. 9: Broadcast route reply

destination. Here the optimal route analysis mode gets the IN node reply as the input, which has the Sequence Number and the HOP number (Fig. 10 and 11).

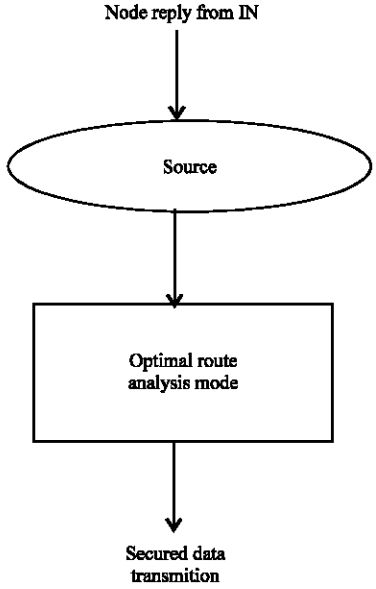


Fig. 10: Optimal path attack preventing

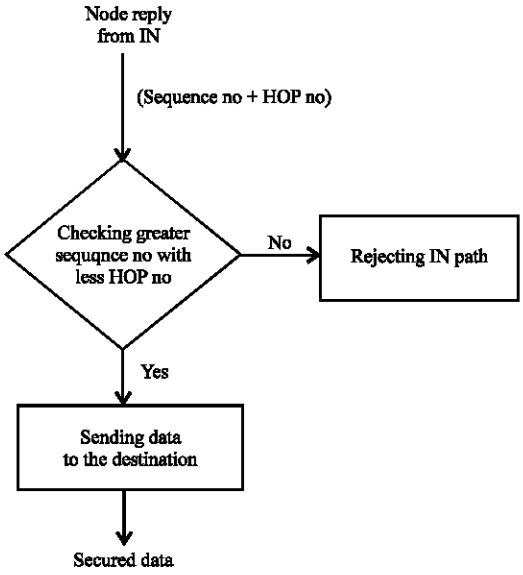


Fig. 11: Optimal route analysis mode

The sequence number is the number of route reply received and the HOP number is the number of terminal crossed in that ad-hoc networks. Now this optimal route analysis mode with its input checks for the validation for the greatest sequence number with the less HOP number. So the destination with this condition is selected as the desired destination and the data packet is decided to send to the destination. If that destination does not have the greatest sequence number and less HOP number, then that intermediate node path is

rejected. So the output of this module is the secured data transmission to the desired destination.

## SIMULATION OF SO2P

**GloMoSim:** Global Mobile Information System Simulator (GloMoSim) is a scalable simulation environment for large wireless and wire line communication networks. GloMoSim uses a parallel discrete-event simulation capability provided by Parsec. GloMoSim simulates networks with up to thousand nodes linked by a heterogeneous communications capability that includes multicast, asymmetric communications using direct satellite broadcasts, multi-hop wireless communications using ad-hoc networking and traditional Internet protocols. The Table 1 lists the GloMoSim models currently available at each of the major layers:

The node aggregation technique is introduced into GloMoSim to give significant benefits to the simulation performance. Initializing each node as a separate entity inherently limits the the scalability because the memory requirements increase dramatically for a model with large number of nodes. With node aggregation, a single entity can simulate several network nodes in the system. Node aggregation technique implies that the number of nodes in the system can be increased while maintaining the same number of entities in the simulation. In GloMoSim, each entity represents a geographical area of the simulation. Hence the network nodes which a particular entity represents are determined by the physical position of the nodes

**Use of GloMoSim simulator:** After successfully installing GloMoSim, a simulation can be started by executing the following command in the BIN subdirectory.
. /glomosim < input file >
The <input file> contains the configuration parameters for the simulation (an example of such file is CONFIG. IN). A file called GLOMO. STAT is produced at the end of the simulation and contains all the statistics generated.

**The visualization tool:** GloMoSim has a visualization tool that is platform independent because it is coded in Java. To initialize the Visualization Tool, we must execute from the *java gui* directory the following: Java GlomoMain. This tool allows to debug and verify models and scenarios; stop, resume and step execution; show packet transmissions, show mobility groups in different colors and show statistics. The radio layer is displayed in the Visualization Tool as follows: When a node transmits a packet, a yellow link is drawn from this node to all nodes within it's power range. As each node receives the packet,

Table 1: Lists the GloMoSim models currently available at each of the major layers

| Layers | Models |
|---|---|
| Physcal (radio propagation) | Free space, two-ray |
| Data link (MAC) | CSMA, MACA, TSMA, 80211 |
| Network (routing) | Bellman-ford, FSR, OSPF, DSR, WRP, LAR, AODV |
| Tansport | TCP, UDP |
| Application | Telnet, FTP |

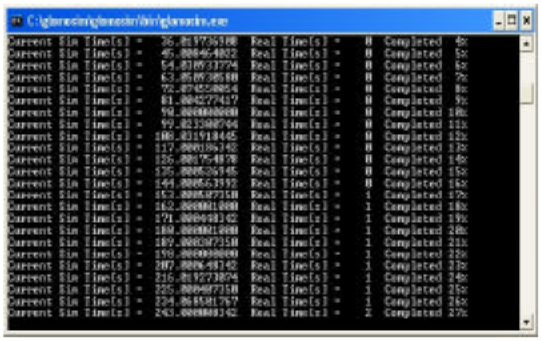

Fig. 12: Simulation of SO2P

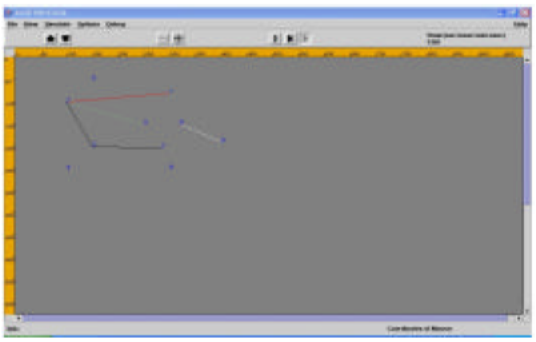

Fig. 13: Visualization tool output

the link is erased and a green line is drawn for successful reception and a red line is drawn for unsuccessful reception. No distinction is made between different packet types (i.e., control packets vs. regular packets, etc) (Fig. 12 and 13)

**Input formats of SO2P routing:**

- From Config File of Glomosim:
- Terrain-dimensions (900, 1000)
- Number-of-nodes 20
- Node-placement File
  - Node-placement-file. /Nodes. Input
- Mobility Trace
  - Mobility-trace-file. /Mobility. in
- Radio-bandwidth 50000

- Routing-protocol so2p
- App-config-file. /cbr. in

**From cbr. in file of GloMoSim**

- CBR 0 6 0 512 3S 40S 170S
- CBR 12 19 0 512 3S 80S 200S
- CBR 24 30 0 512 2S 0S 150S

**From mobility. in file of GloMoSim**

- 2 500MS (350, 50, 0)
- 15 500MS (510, 585, 0)
- 28 500MS (525, 735, 0)

**From nodes. input file of GloMoSim**

| | |
|---|---|
| 0 (100, 200) | 10 (550, 300) |
| 1 (200, 200) | 11 (300, 400) |
| 2 (300, 200) | 12 (150, 500) |
| 3 (400, 200) | 13 (250, 500) |
| 4 (500, 200) | 14 (350, 500) |
| 5 (600, 200) | 15 (450, 500) |
| 6 (700, 200) | 16 (550, 500) |
| 7 (250, 100) | 17 (650, 500) |
| 8 (450, 100) | 18 (750, 500) |
| 9 (350, 300) | 19 (850, 500) |

**Output of data packets delivery in secured on demand position based private routing protocol (SO2P):**

- Node: 2, Layer: RoutingAo2p, Number of Routes Selected = 2
- Node: 2, Layer: RoutingAo2p, Number of Data Txed = 2
- Node: 13, Layer: RoutingAo2p, Number of Data Packets Received = 2
- Node: 19, Layer: RoutingAo2p, Number of Routes Selected = 5
- Node: 19, Layer: RoutingAo2p, Number of Data Txed = 5
- Node: 18, Layer: RoutingAo2p, Number of Data Packets Received = 2
- Node: 11, Layer: RoutingAo2p, Number of Routes Selected = 7
- Node: 11, Layer: RoutingAo2p, Number of Data Txed = 7
- Node: 10, Layer: RoutingAo2p, Number of Data Packets Received = 3

## CONCLUSION

SO2P provides a security of Ad-hoc On Demand Position based private Routing Protocol. It prevents position information from internal and external hackers. It proposes a new secured routing algorithm in mobile Ad-hoc networks and also it finds the optimal path to reach the destination. The DES algorithm is one of the best algorithm to prevent an internal and external attackers and its new approach to the mobile environment proves to be the best in its kind.

In this SO2P made some difficulties in the performance of routing because of Security Mechanisms. E. g. High packet loss ratio, less throughput, More time in end to end connection delay. So In my future work I will aim to improve the performance of SO2P using Backup Routing mechanism. This mechanism provide a sophisticated Back-up routing protocol for mobile Ad-hoc network in case of routing path failure occurred in networks, we also ensure to increment the performance of SO2P protocol

## REFERENCES

Asokan, N. and P. Ginzboorg, 2000. Key agreement in Ad-hoc networks. Computer Commun. Rev., 23: 1627-1637.

Broch, J., D.A. Maltz, D.B. Johnson, Y.C. Hu and J. Jetcheva, 1998. A performance comparison of multi-hop wireless ad hoc network routing protocols. In Proceedings of the Fourth Annual International Conference on mobile Computing and Networking, pp: 85-97.

Cheshire, S. and B. Aboba, 2001. Dynamic configuration of ipv4 link-local addresses. IETF INTERNET DRAFT, zeroconf working group. draft-ietf-zeroconf-ipv4-linklocal-03. txt.

Cheung, S., 1997. An efficient message authentication scheme for link state routing. In: 13th Annual Computer Security Applications Conf., pp: 90-98.

Dahill, B., B.N. Levine, E. Royer and C. Shields, 2001. A secure routing protocol for ad hoc networks. Technical Report UM-CS-2001-037, university of Massachusett, Departament of Computer Science.

Guerrero, M., 2001. Secure ad hoc on-demand distance Vector (SAODV) routing. IETF MANET Mailing List, Message-ID C17B40. BBF52E09@nokia. com, http://www. cs. ucsb. edu/ eroyer/txt/saodv. txt.

Haas, Z.J., M.R. Pearlman and P. Samar, 2002. The interzone routing protocol (IERP) for ad hoc networks. INTERNET DRAFT, MANET working group, draft-ietf-manet-zone-ierp-02. txt.

Hauser, R., A. Przygienda and G. Tsudik, 1997. Reducing the cost of security in link state routing. In Symposium on Network and Distributed Systems Security (NDSS '97), California, Internet Society. pp: 93-99.

Hu, Y.C., D. Johnson and A. Perrig, 2002. SEAD: Secure efficient distance vector routing for mobile wireless adhoc networks. In Fourth IEEE Workshop on mobile Computing Systems and Applications (WMCSA '02), pp: 3-13.

Hu, Y.C., A. Perrig and D. Johnson, 2001. Ariadne: A secure on-demand routing protocol for ad hoc networks. Technical Report TR01-383, Rice University.

Internet Society, 1997. Symposium on Network and Distributed Systems Security NDSS '97San Diego.

Johnson, D.B. *et al*., 2002. The dynamic source routing protocol for mobile ad hoc networks (DSR). INTERNET DRAFT, MANET working group, draft-ietf-manet-dsr-07. txt.

Kent, S., C. Lynn, J. Mikkelson and K. Seo, 2000. Secure border gateway protocol (S-BGP)-real world performance and deployment issues.

Krawczyk, H., 2000. Simple forward-secure signatures from any signature scheme. In ACM Conference on Computer and Communications Security, pp: 108-115.

Madson, C. and R. Glenn, 1998. The use of HMAC-MD5-96 within ESP and AH. Internet Request for Comment RFC 2403.

Madson, C. and R. Glenn, 1998. The use of HMAC-SHA-1-96 within ESP and AH. Internet Request for Comment RFC 2404.

Marti, S., T.J. Giuli, K. Lai and M. Baker, 2000. Mitigating routing misbehavior in mobile ad hoc Networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking, pp: 255-265.

Montenegro, G. and C. Castelluccia, 2002. Statistically Unique and Cryptographically Verifiable (SUCV) identifiers and addresses. Network and Distributed System Security Symposium (NDSS '02).

O'Shea, G. and M. Roe, 2001. Child-proof authentication for mipv6, ACM Computer Communication Review.

Papadimitratos, P. and Z.J. Haas, 2002. Secure routing for mobile ad hoc networks. SCS Communication Networks and Distributed Systems Modeling and Simulation Conference.

Perkins, C.E., E.M. Royer and S.R. Das, 2002. Ad hoc On-demand Distance Vector (AODV) routing. IETFINTERNET DRAFT, MANET working group, draft-ietf-manet-aodv-10. txt.

Perlman, R., 1983. Fault-tolerant broadcast of routing information. In Computer Networks, pp: 395-405.

Perrig, A., R. Canetti, D. Song and D. Tygar, 2001. Efficient and secure source authentication for Multicast. In Network and Distributed System Security Symposium (NDSS'01).

Perrig, A., R. Szewczyk, V. Wen, D.E. Culler and J.D. Tygar, 2001. SPINS: security protocols for sensor networks. In Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking, pp: 189-199.

Ramanathan, S. and M. Steenstrup, 1996. A survey of Routing techniques for mobile communications networks. Mobile Networks and Applications, 1: 89-104.

Royer, E.M. and C.-K. Toh, 1999. A review of current routing protocols for ad hoc mobile wireless networks. IEEE. Personal Commun., pp: 46-55.

Smith, B.R., S. Murthy and J.J. Garcia-Luna-Aceves. Securing distance-vector routing protocols. In: Symposium on Network and Distributed Systems Security (NDSS '97) pp: 85-92.

Stajano, F. and R. Anderson, 1999. The resurrecting Duckling: Security issues for ad-hoc wireless networks. In Proceedings of the 7th International Workshop on Security Protocols, number 1796 in Lecture Notes in Computer Science, Springer-Verlag, Berlin Germany, pp: 172-194.

Thomson, S. and T. Narten, 1998. Ipv6 stateless address auto configuration. IETF Request for Comments, RFC 2462.

Zhang, K., 2001. Efficient protocols for signing routing messages. In: Proceedings of the Symposium on Network and Distributed Systems Security (NDSS'98).

Zhou, L. and Z.J. Haas, 1999. Securing ad hoc networks. IEEE. Network Mag., 13: 24-30.