Journal of Computer Science 7 (6): 924-936, 2011
ISSN 1549-3636

A Secure Routing Protocol to Eliminate Integrity, Authentication and Sleep Deprivation Based Threats in Mobile Ad hoc Network

Edna Elizabeth Nallathambi, Radha Sankararajan,
Vaithiyanathan Sundaram and Gracelin Sheeba
Department of Electronics and Communication, SSN College of Engineering,
Kalavakkam, SSN Nagar, Old Mahapalipuram Road, Chennai, 603110, India

Abstract: Problem statement: Network security in Mobile Ad hoc Network (MANET) is a major issue. Some of the attacks such as modification, impersonation, Time To Live (TTL) and sleep deprivation are due to misbehaviour of malicious nodes, which disrupts the transmission. Some of the existing security protocols such as ARAN, SAODV and SEAD are basically used to detect and eliminate one or two types of attacks. The major requirement of a secure protocol is to prevent and eliminate many attacks simultaneously which will make the MANETs more secured. **Approach:** We propose the algorithm that can prevent and also eliminate multiple attacks simultaneously, called MIST algorithm (Modification, Impersonation, Sleep deprivation and TTL attacks). This algorithm is written on Node Transition Probability (NTP) based protocol which provides maximum utilization of bandwidth during heavy traffic with less overhead. Thus this has been named MIST NTP. **Results:** The proposed MIST NTP has been compared with NTP without the MIST algorithm, Authenticated Routing for Ad hoc Networks (ARAN) and Ad hoc on Demand Distance Vector (AODV). Extensive packet level simulations show that MIST NTP produces around 10% less end to end delay than ARAN, it even drops 30% fewer packets compared to malicious NTP on an average and around 50-60% fewer packets compared to AODV during multiple attacks. **Conclusion:** The results ensure that MIST NTP can break the greatest security challenge prevailing in MANETs by securing the MANET against several attacks at once.

Key words: Mobile Ad hoc Network (MANET), Ad hoc on Demand Distance Vector (AODV), Authenticated Routing for Ad hoc Networks (ARAN), routing protocol, security cryptography, Flooding Attack Prevention (FAP), Node Transition Probability (NTP)

INTRODUCTION

Ad hoc network is a very popular wireless networking paradigm for mobile hosts. MANET (Mobile Ad hoc Network) is a collection of communication devices or nodes that wish to communicate without any fixed infrastructure and pre-determined organization of available links. In general, any wireless network is highly vulnerable to security attacks and dealing with this is one of the main challenges of developers of these networks today.

Several popular Ad hoc routing protocols have been addressed for securing the MANET against different types of attacks, such as modification, impersonation and fabrication (Bing *et al*., 2006; Diffie and Hellman, 1976). In this study, the above mentioned attacks are examined under the headings such as availability, confidentiality, authentication and integrity. A network is considered to be safe when all these four concerns are considered and are safe against the threats against these.

Availability refers to the fact that the network must remain operational at all times despite denial of service attacks. In this study, it is assumed that the source itself acts as the central authority and hence keys are available whenever required. Attacks such as denial of service can also be caused due to "sleep deprivation attack" and this is prevented and eliminated by a Self recovery algorithm which is a part of MIST algorithm implemented on NTP. Confidentiality ensures that certain information is never disclosed to certain users. This study shows that the problem of impersonation attack can be eliminated by the identity based method of key generation. Thus, the authentication algorithm provides defence for "Impersonation attack" thereby authenticating the source and the destination.

Corresponding Author: Edna Elizabeth Nallathambi, Department of Electronics and Communication, SSN College of Engineering, Kalavakkam, SSN Nagar, Old Mahapalipuram road, Chennai-603110, India Tel: +91 9444943004 Fax: 044-27474844

Table 1: Vulnerabilities of various protocols

	Types of attacks				
Protocols	Modification attack	Impersonation attack	Sleep attack	Deprivation TTL attack	Black hole attack
TOGBAD (Gerhards-Padilla *et al*., 2007)	No	No	No	No	Yes
Kurosawa and Jamalipour (2007)	No	No	No	No	Yes
Yi *et al*. (2005)	Yes	No	No	No	No
Hu *et al*. (2003)	Yes	No	No	No	No
ARAN (Sanzgiri and Belding, 2002)	No	Yes	No	No	No
Ariadne (Hu *et al*., 2002)	No	Yes	No	No	No
MIST NTP (Proposed)	Yes	Yes	Yes	Yes	Yes

Integrity guarantees that a message is never corrupted when transferred. The modification of data is prevented by the Malicious Detection and Elimination (MDE) method of MIST algorithm, which defends the network against "modification attack" and "TTL attack". Thus, the proposed MIST algorithm is a set of three algorithms (MDE, authentication algorithm and self recovery algorithm) which can efficiently secure the MANET against the four major threats as explained earlier. The complete working of the algorithm has been explained below. Table 1 shows the analysis of various protocols which are vulnerable to many other attacks, though they could secure the network against one threat or attack.

Related works: Generally, during a black-hole attack, a malicious node impersonates a destination node by sending a spoofed route reply packet to a source node that initiates a route discovery thereby creating a threat to the integrity of the network. We have introduced an attack called modification attack, where the malicious attacker changes the intended destination to some un usable destination. This attack can harm the network very badly. Similar to this attack Bo *et al*. (2007) had found an attack called Rushing attack which can also modify the intended destination by making the Request packet to reach the target faster than the request packet from non malicious nodes making the target to think that the malicious route is the intended route. But, in our scenario the malicious node changes the intended destination by sending a wrongly assigned beacon packet to the target node. The MDE algorithm in MIST can completely identify the modification attack and can prevent it. This modification attack can pave way for a malicious node to get control and induce black hole or grayhole attack, which makes its prevention a paramount importance. Kurosawa and Jamalipour (2007) proposed anomaly detection scheme using dynamic training method in which the training data is updated at regular intervals. Their method has claimed to eliminate black hole attack in Ad hoc on Demand distance Vector (AODV) protocol.

Gerhards-Padilla *et al*. (2007) introduced the network with black hole. They devised TOGBAD, a new centralized approach, using topology graphs to identify nodes attempting to create a black hole. They performed plausibility checks of the routing information propagated by the nodes in the network which triggers an alarm if the plausibility check fails.

Moreover, Yi *et al*. (2005) introduced a new Denial Of Service (DOS) attack and its defence in Ad hoc network is called the Ad hoc Flooding Attack. It will exhaust the communication bandwidth and node resource so that valid communication cannot be established which leads to a generic defence against it, called Flooding Attack Prevention (FAP). This algorithm fails if there are other attacks such as sleep deprivation or battery exhaustion which leads to DOS attack in the network.

Bo *et al*. (2007) devised a good defence against the rushing attack. This new attack which results in denial-of-service attack is introduced in this study. Along with the rushing attack they had also developed Rushing Attack Prevention (RAP) which has the ability to eliminate only one type of attack and fails in case of many attacks. Sanzgiri proposed (Sanzgiri and Belding-Royer, 2002) a well known secure routing protocol for Ad hoc network called Authenticated Routing for Ad hoc Network (ARAN). It was based on certificates and was successful in defending the network against all identified attacks to network's authentication. Although, it can authenticate the Ad hoc network very well, it gives lesser performance when used for mobile nodes because of the overhead costs due to mobile nodes sending and receiving signatures. In addition to this, ARAN cannot handle when an attack to authentication of network is combined with a denial of service attack by the same malicious node.

Finally one of the popular and most widely used multiple attack secure routing protocols was Ariadne, proposed by, Bo *et al*. (2007). Ariadne prevents attackers or compromised nodes from tampering with uncompromised routes consisting of uncompromised nodes and also prevents a large number of different types of Denial-of-Service attacks. In addition to that, Ariadne is efficient, using only highly efficient symmetric cryptographic primitives which makes Ariadne the only protocol of those times (to the best of our knowledge) can secure the network from DOS attack and also provides authentication.

Motivation: After studying different types of attacks through literature survey, it is found that there is no single algorithm available to prevent almost all the attacks using a single protocol. This created an important requirement of such protocol. NTP (Node Transition Probability based routing algorithm) (Radha and Shanmugavel, 2003) is the protocol over which our MIST algorithm is written to secure the MANETs against several attacks simultaneously. This NTP based routing algorithm uses the power table which holds the values of the received power of the beacon packet received by the source. It also uses the total number of replies sent by various nodes at a particular energy level which is used to compute the probability of how the network has to route the packet, by utilizing the maximum bandwidth at higher traffic and costing lesser overhead.

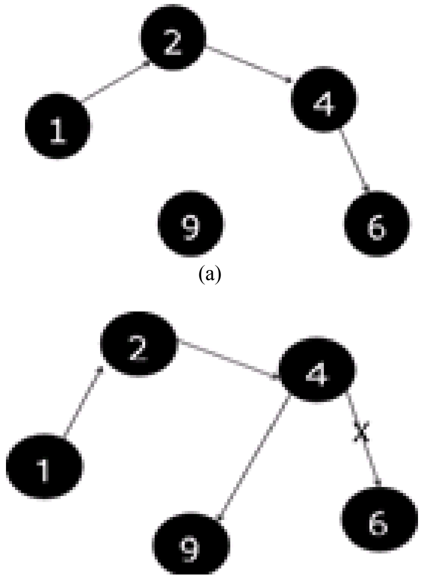
Since this referred NTP routing algorithm was not secured against any threats that are available in MANETs, this study proposes a complete set of separate individual security algorithms which when combined (MIST algorithm) can secure the network against Modification attack, Impersonation attack, Sleep deprivation attack and TTL attack. In this study, the proposed algorithm against security attacks can provide very good performance in terms of good packet delivery ratio, less end-to-end delay.

MATERIALS AND METHODS

The MIST-NTP (Modification, Impersonation, Sleep deprivation and TTL attacks in Node Transition Probability) based routing algorithm establishes a route using its control packets. There are basically two control packets each having its own functions. In brief, the source which wants to establish a connection to a target destination will broadcast beacon packets. Once a node receives the beacon packet, it will check for the destination. If the node that receives the beacon packet is not the intended destination it will forward the packet to its neighbours. During every reception of the beacon packet the node will update its power table with the power level of the beacon packet received. Using the power level it calculates the nearness probability (i.e., the probability that determines how near the particular node is with the destination). Using this probability it establishes the connection between a source and an intended destination. Every malicious attack occurs during this process of establishing a connection. The proposed MIST- NTP will completely secure the network against multiple attacks. As discussed earlier that MIST security algorithm comprises of three algorithms namely, MDE, authentication algorithm and self recovery algorithm. Each individual algorithm in MIST has been explained individually for better understanding of the overall working of MIST-NTP.

Modification attack: Integrity (Stajano and Anderson, 1999) means ensuring that the node has not been maliciously altered. In such cases, when the destination address of a node is tampered or maliciously altered, the node will start sending the packets to the modified destination node instead of the intended destination node. When this happens there can be two fatal consequences: Firstly, the intended node will not receive the packets so as to form a route during the initialization phase and this consequently creates a break in the formation of the wireless topology. Secondly, this could even challenge the privacy of the data in the network, if the malicious node is successful in initiating the attack; the entire network is attacked by the intruder. For convenience in discussions, we call NTP as Malicious NTP (MNTP) when it gets attacked by modification attack.

Figure 1a-b illustrate the actual path that has to be followed and the path after malicious activity taken place respectively. The intended route from Fig. 1a is 1-2-4-6 but due to malicious activity the destination is changed to 9 and thus making the route 1-2-4-9. The data is sent to node 9 instead of node 6 thereby the data is lost due to lack of authenticity.



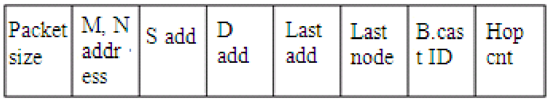
(a)

(b)

Fig. 1: Before (a) and after (b) malicious activity

Malicious node detection and elimination (MDE): For detection and elimination of the malicious node which causes modification attack, a novel algorithm named as Malicious node Detection and Elimination (MDE) is used, which uses the beacon control packet of NTP protocol. As mentioned earlier this is one of the algorithms in MIST. The following are the steps implemented in MDE algorithm to detect and eliminate the malicious node in the network (Radha *et al*., 2010):

- The MN address field is added to the beacon packet assuming that it is a non-mutable field and hence the information can be only updated and not altered. Also the last address field in the beacon packet is assumed to be a non-mutable field
- When the node has received the beacon packet, it will compare the same with other beacon packets that it had received from its other neighbours. Using these received beacon packets, the node checks for any changes in the destination address of the packets it has received. The node also checks the flood number in the packet, for finding if there is really a malicious activity or if this is just a packet during connection establishment between the same source and a different destination. The new route discovery will have its flood number starting from one, whereas the maliciously altered packet will have higher flood. This checking avoids false positives
- If the destination address of any one of the beacon packets has been changed by the malicious node then the node on receiving the beacon packet will find the last address field (address of the previous node), from where the beacon packet has been received and updates the MN address field with the malicious node address in the last address field, which is shown Fig. 2
- On receiving the beacon packets, from neighbouring nodes, the nodes will update their neighbour tables by flushing the node corresponding to the MN address field, thereby eliminating the malicious node from the network. The use of a separate packet to notify the nodes about the malicious node has been avoided by using an additional field (MN address) in the beacon packet itself. This costs comparatively less overhead than the usage of a separate packet to notify in the network



Packet size	M, N address	S add	D add	Last add	Last node	B.cast ID	Hop cnt

Fig. 2: Beacon packet with MN address field

Impersonation attack: We generally come across nodes spoofing the identity of other nodes. This type of attack is called as impersonation attack. This occurs when the malicious node cloaks just like the intended destination and will extract all the resources from the packet it receives by impersonating like the actual node. If a malicious node is inducing both modification attack and attacks the authenticity of the network as well, then only the MIST algorithm can handle those situations. To avoid the attack to authenticity or to avoid impersonation attack, we provide the second part of MIST algorithm that provides authentication between the source and destination by implementing a method of key exchange in the network layer as exposed in our research (Radha *et al*., 2010).

Authentication algorithm: Authentication is provided based on the node ID and it has an ID based security scheme. Authentication is assured between source and destination node by implementing a key exchange and cryptographic algorithms in search of control packet of NTP protocol. Basically, the Authenticated Routing for Ad hoc Networks (ARAN) protocol uses secure certificates to authenticate the network so as to prevent most of the threats to the authenticity like cache poisoning, tunnelling, impersonation. In the proposed algorithm instead of using certificates from a trusted certificate server we use public keys and private keys generated by the KGC using the ID (Identity) of the individual nodes. It is assumed that each source will act as Key Generation Centre (KGC). It is also assumed that encryption algorithm is known only to source and destination nodes. Source creates public keys and private keys using the ID (Identity) of the individual nodes and sends those keys through the secure channel (Ren *et al*., 2007). Secure channel is established by sending all the keys encrypted through the search packet which is the control packet of an existing NTP routing algorithm. Separate control packets for sending the keys are avoided in order to reduce the control overhead in the network:

- Let P node be public key of node
- Let K (P node) be the private key of the node

As per Diffie Hellmann (Diffie and Hellman, 1976) key exchange method, source creates an arbitrary key for source and destination nodes and also creates a common key using private key of source and destination nodes using the following steps.

$$Y_s = (\alpha)^{ks} \quad (modq)$$
$$Y_d = (\alpha)^{kd} \quad (modq)$$

$$K_{cs} = (Y_d)^{ks} \ (\text{modq})$$
$$K_{cd} = (Y_s)^{kd} \ (\text{mod } q)$$

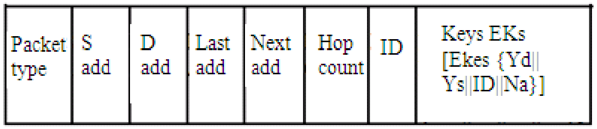
Where:

K_s	=	Private key of source
K_d	=	Private key of destination
K_{cs}	=	Common key generated by source
K_{cd}	=	Common key generated by destination
Y_s	=	Arbitrary source key required to generate the common key by destination
Y_d	=	Arbitrary destination key required to generate the common key by source
ID_s	=	Identity of source
N_a	=	Nonce of source A
P_s	=	Public key of source

Then source concatenates the arbitrary key of source (Y_s) and destination (Y_d), ID of source (ID_s) and nonce (N_a). After concatenation of these keys, it encrypts them with the common key (k_{cs}) using simple symmetric algorithm and this is further encrypted with source private key using RSA algorithm (Rivest *et al*., 1978). These generated keys are added in a separate field of the search packet and then transmitted.

After receiving the search packet at destination node (the search packet with the additional authenticating field is shown in Fig. 3), the above mentioned keys are decrypted using public key of source (P_s) and common key "k_{cs}", then the values are de-concatenated to retrieve the arbitrary key of source node, arbitrary key of destination node, ID of source (ID_s) and nonce (N_a). Now the destination calculates the arbitrary key (Y_d) using its private key "k_d" and common key "k_{cd}" and it verifies if "k_{cs}" is equal to "k_{cd}". If they are equal it authenticates and ensures that the control packet is sent by the intended source. If "k_{cs}" is not equal to "k_{cd}", it makes the node get alerted that there is some malicious activity.

Such malicious node can be detected by variations in delay that is caused by processing (decryption) and computation of the key. This method helps to detect the malicious nodes and eliminate them from the network. This method is expected to provide two advantages; one is increase in number of packets received by destination, when authentication is assured and another advantage is decrease in delay that is caused by malicious node. The only disadvantage is increase of control overhead when compared to ARAN (Sanzgiri and Belding-Royer, 2002) because more number of keys is needed to be added with the packets as shown in Fig. 3. But, as proved in the results section MIST algorithm gets better than ARAN in spite of this little increase in control overhead when there are fast moving nodes.



Packet type	S add	D add	Last add	Next add	Hop count	ID	Keys EKs [Ekes {Yd||Ys||ID||Na}]

Fig. 3: Search control packet

Detection and elimination of impersonation attack: When the malicious node, replaces its identity with that of the source, impersonation attack occurs. The malicious node tries to find the Identity (ID) of the source in the beacon key field and replaces by its own identity in that field. So when the keys are exchanged and if it is found that "k_{cs}" is not equal to "k_{cd}", it is understood that the impersonation attack has taken place. Due to this attack a delay is encountered in the network. The last address field content is updated in the routing table and this node is considered to be the malicious node. The delay is due to the processing delay (change of ID and decryption of keys) in the beacon packet. Thus the authentication algorithm in the proposed MIST algorithm can notify the other nodes by updating the MN address field in the beacon packet. Once the beacon is broadcasted all the neighbours who receive this beacon can flush the malicious node from their routing table. This shows the coexistence of the MDE and authentication algorithm in MIST. This makes the MIST algorithm a good security algorithm against multiple attacks at the same time. Table 2 shows the overall algorithm.

Sleep deprivation attack (or) Sleep deprivation torture attack: The first two parts of the MIST algorithm can defend the network against authentication, integrity based threats. The next most important and highly sensitive are Denial of Service attacks which makes the network to even cease in the worst case. Often in real time scenarios we come across a major drain in the battery life of the node that is being used. When such a situation occurs, the node will stop sending packets and hence the information which is supposed to transmit, will be lost. In another case, the node does not shut down and becomes selfish to retain its battery life and hence starts denying packets leading to a Denial of Service Attack and Packet dropping. This ultimately leads to loss of data. Thus in this case it is absolutely essential to keep track on the battery life of the nodes in the network and to conserve their battery life once it falls below an optimum threshold level.

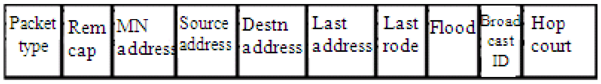
Referring (Vaithiyanathan *et al*., 2010a; 2010b) Stajano and Anderson state that battery exhaustion when left unnoticed will be a threat to the availability for the users getting benefitted by the network.

Table 2: Authentication algorithm in MIST for defending impersonation attack

Step 1:	ID-identity of the individual node is given to Key Generation Centre (KGC)
Step 2:	The source acts as the KGC centre
Step 3:	The source creates a public key and private key for every node which sends its ID to it
Step 4:	Keys are exchanged using diffie hellman process
Step 5:	The keys are sent through a secured channel
Step 6:	The search control packet is used as the secured channel for the keys. This ensures that there is no chance of any stealing of keys as in the case of Man In The Middle Attack (MITM).
Step 7:	The keys are encrypted and decrypted using RSA algorithm (Rivest *et al*., 1978)
Step 8:	The common keys are verified for authentication. If they are equal then the source and destination are authenticated
Step 9:	If they are not equal, impersonation attack is detected.
Step 10:	The algorithm once it detects the presence of some malicious activity due to which the network is threatened against its authenticity, it will trace the last address in the received search packet.
Step 11:	It will then update the MN address field of the beacon packet and notifies the neighbours broadcast that. The process of flushing the malicious node from the routing table will be taken care by the MDE algorithm which was explained in the first part of the MIST algorithm.

Also, a malicious node taking advantage of this "dying node" (we mention "dying node" when a node is almost out of its battery capacity and will be completely dead in some time due to its battery exhaustion) can cause a sleep deprivation torture attack which is one of the possibility to induce Denial of Service attack. Understanding the critical importance of battery life this MIST algorithm which runs in network layer can help the network to get notified when any of its nodes is under the danger of dying due to less battery life. The explanations below, explains how the self recovery algorithm (third part of MIST algorithm) can prevent sleep deprivation attack in addition to the other attacks explained earlier.

Self recovery algorithm and "dying out node" tracking algorithm in MIST: The first phase in our self recovery algorithm is to detect or track the node that is having the battery capacity depleting below the threshold level. This phase of tracking the "dying out node" is called the "Dying out node" tracking method. As mentioned above, the "dying out" node is a node that has its battery capacity depleting below the threshold value (30% of the total battery capacity-the reason for using 30% is explained in discussions). Once the source node initiates a connection establishment request with a destination, every node sends the beacon along with the RemCap value. This value in the non mutable field denotes the remaining battery capacity. The battery capacity of a node can be extracted using the ACPI (Advanced Configuration Power Interface) standard which is an open standard that interfaces between the battery level of the node and the operating system. This open source is available for both linux/unix and windows platforms. Though we have simulated the whole setup, this ACPI can be used in real time implementation to extract the battery level of the node and, the remaining capacity can be calculated in mAH before updating the RemCap field.

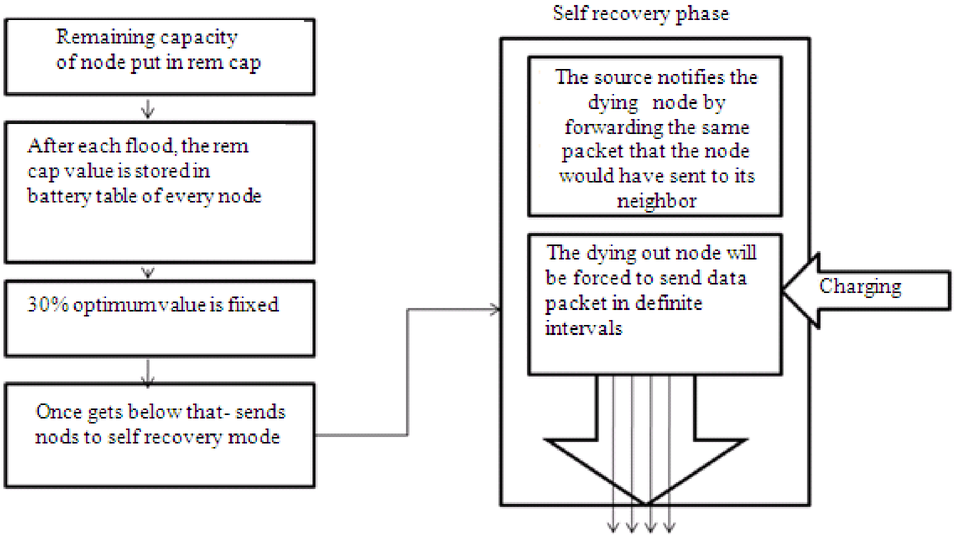


Packet type	Rem cap	MN address	Source address	Destn address	Last address	Last rode	Flood	Broad cast ID	Hop court

Fig. 4: Beacon packet with rem cap field

Initially, MN address field will be filled if there is a modification attack occurred and other nodes will learn about the malicious node using this or, if the network is not affected by malicious activity the MN address field will be NULL. Once, the node receives the beacon (Fig. 4) from other nodes, the remaining battery capacity value is stored in a table called "battery table" (similar to routing table, neighbour table this will also be stored in the router cache). The neighbours regularly check their battery table for the battery capacity of its neighbours to be over the threshold capacity. This ensures that all the neighbours learn about its neighbours within its range in case of depletion of battery capacity below the threshold value and it notifies the dying node to get into self recovery mode. The total battery capacity is 2800mA-hr (This is the maximum capacity of a Li Dell E4200 battery).

Figure 5 shows the process before and after a node getting into the self recovery phase from "dying out node" tracking phase, when the battery capacity of the node dips the threshold level, the neighbours immediately notify the dying node and sends it to the recovery phase. The notification to dying node is done by forwarding the same packet that the node would have sent to its neighbour and when the receiving node finds that the packet it has received is same as the one it had sent, it will know that it has to enter into the self recovery phase. All the nodes' API has been built such that the node switches itself into self recovery mode if it gets its own packet sent by other node. Once the route is established, each node will send some number of data packets to the destination giving rise to the reduction in the remaining battery capacity. The source updates the battery table to track the optimum threshold value.


Remaining capacity of node put in rem cap
After each flood, the rem cap value is stored in battery table of every node
30% optimum value is fiixed
Once gets below that- sends nods to self recovery mode
Self recovery phase
The source notifies the dying node by forwarding the same packet that the node would have sent to its neighbor
The dying out node will be forced to send data packet in definite intervals
Charging

Fig, 5: Diagrammatic representation of sleep deprivation attack elimination

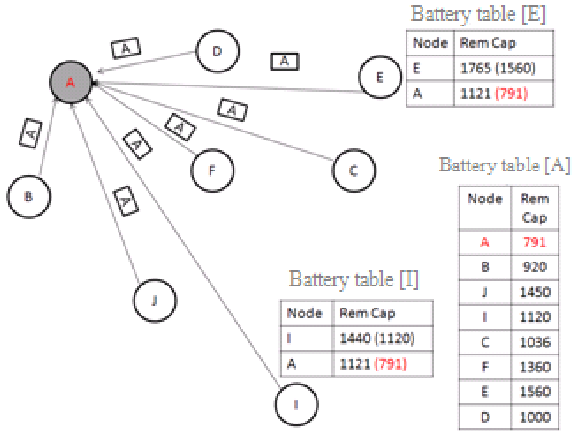
The sleep deprivation attack with an eight node scenario is explained in Fig. 6-7. These nodes are named alphabetically from A-J and are placed randomly in a rectangular grid. Every node is considered to be active and the operation of node A is highlighted. Node A updates its battery table once it receives the packets from the other nodes. Once the battery table is not empty, the nodes starts to check if there is any node that has battery level (RemCap) lesser than the threshold value.

Figure 6 shows the situation when Node A (represented in red) consumes its available battery capacity and dips below the threshold level. The value shown inside the brackets is the battery capacity updated after the occurring flood and the value outside the brackets represent the battery capacity during the previous flood. Initially, all the other nodes are alerted with the help of the battery table, that, Node A's battery capacity is below the threshold level. Then, all the neighbouring nodes will forward the packet that was originally sent by A, to node A. Node A on receiving its own packet sent by other nodes switches into self recovery node.

In the self recovery phase the node sends the data packets at regular intervals rather than sending it continuously. If the node generally sends four packets every second, during the self recovery phase it sends one packet every quarter of a second until its battery gets charged back to the normal level over the threshold. There will be a little delay for the packet to reach the destination as the packets are sent at intervals. But, this little compromise on delay is better than making the node work in its full efficiency and stopping it from sending any packet once it dies out. The operation of the self recovery mode is shown in Fig. 7. The alerted node A will send the data packets (not the beacon packets) at regular intervals; so that it can preserve its battery capacity, which once recharged (it is assumed that the node can either be recharged manually once any user gets notified about the depletion of battery capacity or can dismantle this node from the network and install a new node instead of that) will get back to normal stage.

Time To Live (TTL) attack: The Time to live parameter (TTL) decides the total time for two nodes to live in the network before establishing a route. Once the TTL expires, the two nodes will stop its communication of control packets. Thus the TTL value should be selected as an optimum value that allows the right amount of control packets to be sent in order, to create a valid route in between nodes such that the total overhead due to these control packets is at an optimized rate. If the TTL value is too high, then the overall number of control packets sent for establishing the route will be increased drastically. If the value is too low, the required number of control packets will not be sent, thus breaking the already existing link or causing no route to be established.

We can observe that the value of the TTL is one of the most important parameters that govern the network. TTL expiry attack or simply TTL attack have attacked wired networks and has a severe impact on the overhead because of the increase in the production of exception packets. This particular attack have been explained in Cisco intelligent security articles and they use access lists (a technique used to control specific inbound packets or networks) to control the TTL attack in their networks.



Battery table [E]

Node	Rem Cap
E	1765 (1560)
A	1121 (791)

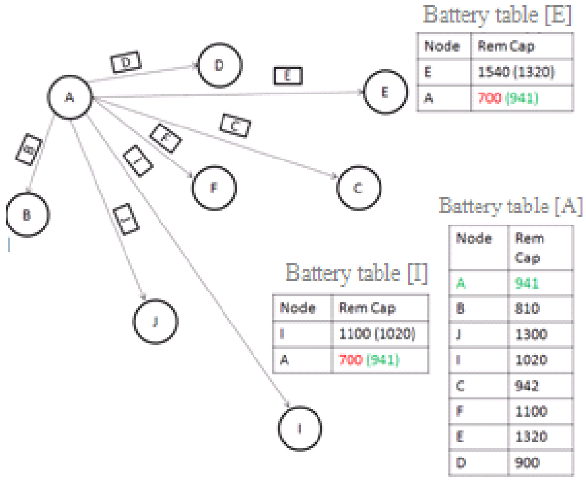
Battery table [A]

Node	Rem Cap
A	791
B	920
J	1450
I	1120
C	1036
F	1360
E	1560
D	1000

Battery table [I]

Node	Rem Cap
I	1440 (1120)
A	1121 (791)

Fig. 6: Behaviour and battery level of node a below threshold



Battery table [E]

Node	Rem Cap
E	1540 (1320)
A	700 (941)

Battery table [A]

Node	Rem Cap
A	941
B	810
J	1300
I	1020
C	942
F	1100
E	1320
D	900

Battery table [I]

Node	Rem Cap
I	1100 (1020)
A	700 (941)

Fig. 7: Behaviour and battery level of node a after recovery

We forecasted the same problem in mobile ad hoc networks where in TTL could be even more serious threat to the increase in control packets. Our MIST algorithm can also control the TTL attack in addition to all the attacks as explained above.

Generally, the malicious attack on TTL occurs when a malicious node requests a connection establishment as a process shown in modification attack earlier, forces the control packets to increase by raising the TTL value during the connection establishment. So, the innocent control packets will be sent until the evil TTL expires. In some cases malicious activity invades a working node and makes the TTL lesser to break any connection establishments request that comes to that node. The consequences of change in TTL value and the elimination processes are explained below.

Elimination of TTL attack using aspects of NTP and MDE algorithm of MIST: The TTL value will be default value set as per the requirement for the network initially and the nodes will establish routes using the beacon packets. If the TTL value is maintained without malicious activity, such that the total time taken by two nodes in order to establish the route is equal to the total time taken for the actual number of beacon packets required for establishing the route, then there is no threat for the network. If malicious activity occurs and if the TTL value is changed to a higher value, then the time taken for establishing the route will be more and it also demands more number of beacon packets for establishing the route. This increases the end to end delay as well. If the TTL value is decreased to a much lesser value then no route will be established. This TTL attack can be defended by the MIST algorithm. It uses the MDE algorithm of the MIST and also some of the aspects of the natural ability of NTP routing algorithm over which this MIST has been implemented. The algorithm shows the process of elimination and prevention of TTL attack over MANETs by the MIST algorithm is shown in Table 3.

Implementation of MIST NTP: The MIST NTP can defend multiple attacks at the same time. To show that we have created the scenarios such that there are malicious activities in the network and the proposed MIST algorithm can detect, eliminate and further prevent those multiple attacks. The same node hasn't been attacked by the multiple attacks discussed here, though that can also be a case which can be detected and eliminated by MIST algorithm. For, more clear picture we induced multiple attacks on different nodes but all at the same time. This gives the algorithm an upper hand as this avoids multiple attacks at the same time irrespective of the attack being on a single node or multiple nodes.

Packet format: In this proposed method, beacon packets of 19 bytes are used. It has both the MN address field and the Rem Cap field in the frame.

Though in the above sections, the Rem Cap field and the MN address field are given separately; during unification of all the individual security algorithms, every beacon control packet will have both the fields in the packet frame. The address in the packet uses IPV4 standard of length 4 bytes each. The size search packet depends on the keys it carries in its payload.

Table 3: Elimination of TTL attack-algorithm

Step 1: The Node Transition Probability Routing Algorithm tries to create a route that costs lesser overhead, so that, if the TTL value changes, the routing algorithm will skip the prevailing route and switch to the next route, thereby changing the destination.

Step 2: If there is a change in the route, the neighbouring nodes would sense that change in the destination before the new destination actually gets updated in the routing table.

Step 3: Thus the step 2 leads to the modification attack. The MDE algorithm of MIST as explained before will detect an anomaly behaviour which, eventually traces the last node from the MN address field. Now the network can detect the malicious node. The neighbours can now flush the malicious node from their route table. During the next flooding, the TTL value can also be reset in the node and can be used again in the network. This part of MIST algorithm can enhance the availability of nodes in the network

RESULTS

The protocols are simulated using GloMoSim library. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation capability provided by PARSEC. We have considered a network scenario with 25 mobile Ad hoc nodes distributed uniformly at random in a 1000×1000 square meters grid, each equipped with 802.11a radios. The link data rate is set to 11Mbps and the radio signal obeys free space propagation with a transmission range of 200 meters. All the nodes obey Random Way Point Model for its mobility. The simulated time interval is1000 seconds and all the data that is gathered were averaged for at least 10 runs. The node speed was varied from 0-800 m sec^{-1}, each having its own significance. There are 3 categories of variations, low speed nodes, medium speed nodes and high speed nodes. All the nodes that move between 050, 50-300 and > 300 m sec^{-1} are assumed as low, medium and high speed respectively. The CBR rate is set to 100 Kbps.

The performance results from Fig. 8-11 explain how the network reacts with and without the MIST algorithm when there are multiple malicious activities attacking the network. These performance results compare the packet delivery ratio, end to end delay, overhead due to control packets and analyses the battery capacity during the malicious attack.

In all the comparison charts, the MIST denotes the proposed MIST algorithm that fairs well in all occasions against multiple attacks at the same time. The MNTP denotes the NTP routing algorithm without MIST algorithm in it. Also, the NTP shows the network performance for the network that is free of any malicious behaviour and MNTP (Malicious NTP) which shows the performance with malicious attacks. The NTP is taken as a comparison in the performance results to show how well the proposed MIST algorithm bring the network attacked by multiple attacks back into track such that it has the performance almost closer to the network without any malicious behaviour.

Multiple attacks have been induced in the network to check the performance of MIST algorithm. Out of the 25 mobile Ad hoc nodes, node no. 12 is attacked by modification attack, node no. 15 is attacked by TTL attack and the node with the least battery (node 1) capacity is selected to induce a sleep deprivation torture attack. GloMoSim doesn't have resources to support network modelling that helps to create KGS and an eavesdropper to eavesdrop and induce an attack to the authentication like impersonation attack. So, we made a node to change the key that is being sent to the destination, such that the destination fails the authenticity. Thus, there are four threats in the network and the following performance results show how the proposed MIST algorithm can defend the multiple attacks simultaneously.

Figure 8 shows that the packet delivery ratio of the MIST NTP is far better than the Malicious NTP (MNTP) because the malicious attacks will reduce the packets reaching the destination. In addition to that having multiple attacks at the same time destroys the whole network which reflects in the packet delivery ratio. The total number of packets delivered during the start with the nodes moving at lowest speed is just 200-250. This shows that there is a 75% decrease in packets delivered due to the network affected by multiple attacks. The NTP is shown in the results to illustrate the ability of MIST NTP to make the network completely affected by multiple attacks to be almost like a network that is not affected by any attack.

Similarly Fig. 9 shows the comparison of End to End delay for nodes moving at speeds of 10-100 m s^{-1}. It can be seen at certain high speeds the MNTP is reacting too slowly, such that the end to end delay drastically increases, but the MIST NTP can act even better than NTP at certain situations. We observe that the reason behind the reduction in end to end delay beyond the ideal NTP without malicious activity is because of the self recovery algorithm of MIST, which makes the node to get notified by the neighbours when the battery level dips beyond the threshold even when there is no malicious activity which marks as one of the best advantage of MIST NTP.

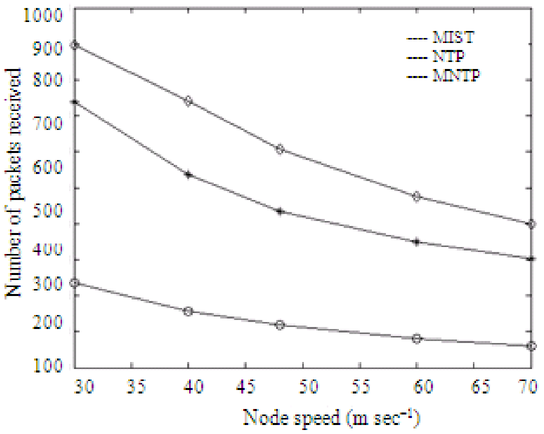


Fig. 8: Number of packets received by destination and node speed

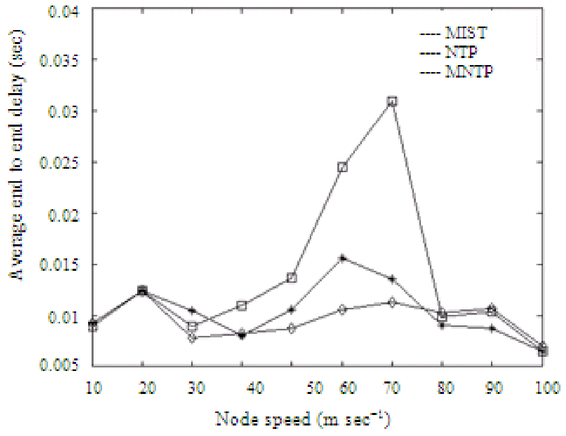


Fig. 9: Average end to end delay Vs node speed

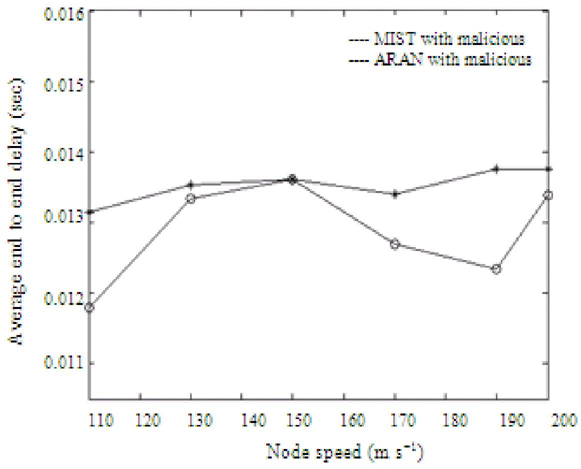


Fig. 10: Comparison of SNTP and ARAN in terms of average delay

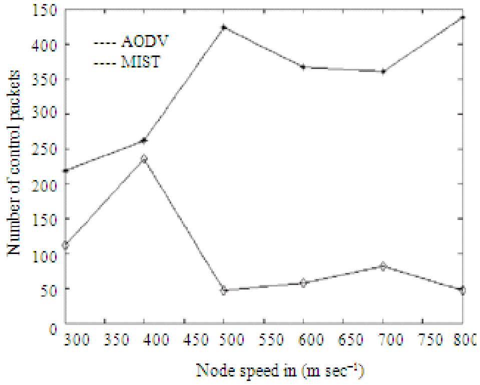


Fig. 11: Control overhead and node speed

Figure 10 shows a specific comparison with MIST and ARAN. This compares the end to end delay for medium speed nodes for the same malicious network which we are using for all the performance measurements but, as it is well known that ARAN is a secure protocol that can authenticate the whole network using its signatures and certificates, we avoided attacking the network with other attacks and attacked only the authenticity of the system to avoid obvious increase in delay. Even then we can observe that the production of certificates in the network creates some amount of delay in the network which increases the end to end delay than the delay produced by MIST. This highlights another advantage of MIST NTP.

Figure 11 compares the MIST NTP with AODV for measuring the overhead increase due to the raise in the control packets. The major raise in control packets are due to the TTL attack that severely attacks the network. Though AODV and MIST NTP starts with almost same overhead, when the network works with TTL attack that is unnoticed it will lead to a drastic raise in the total control packets involved before establishing connection. To be more precise the measurements in Fig. 11 were done in a network where TTL attack dominates other attacks, as the major motive for this performance measurement is to find the variation in the control packets used for connection establishment in MIST NTP and AODV (which is a well known routing protocol that establishes connection with the help of control packets).

DISCUSSION

Setting the threshold value for self recovery algorithm: As analysed by (Khan *et al*., 2008) in the threshold value for a network to be safe from unstable nodes (i.e., nodes with low remaining battery capacity which will be eventually be deprived of battery life and will start denying its requests), the battery threshold value should be set to 30% of the total battery capacity.

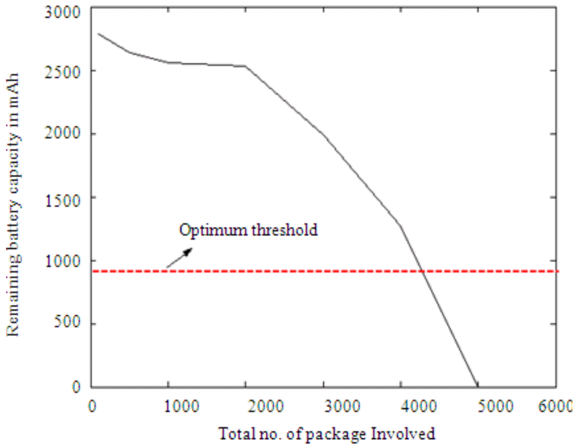


Fig. 12: Analysis of stability of node 1 at different quantity of packets to send

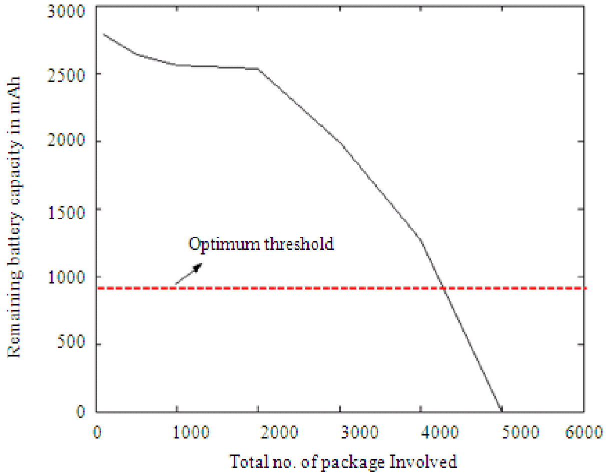


Fig. 13: Analysis of stability of node 1 at different node speeds

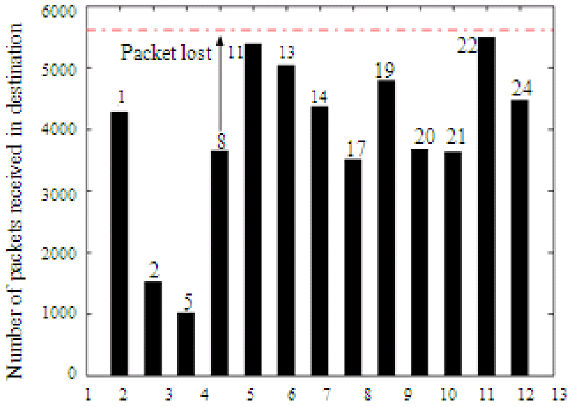


Fig. 14: Analysis of impact due to died out nodes

Their results in (Khan *et al*., 2008) ensure the threshold's correctness which is supported by our results from Fig. 11-14. The performance study of AODV in (Khatri *et al*., 2010) helps us to compare our performance results with AODV with improved results

Analysis of sleep deprivation attack and the battery consumption without malicious activity: As examined by Wander *et al*. (2005) 59.2 micro joules of energy will be consumed for sending one packet of 1 byte length. Here a 512 bytes data packet is used.

Gradually, the remaining battery capacity of the node gets reduced as each and every node starts transmitting data packets and the other nodes keep receiving. Also as mentioned in (Wander *et al*., 2005), energy will be consumed not only when the packet is transmitted but also when it is received.

That is 28.6 micro joules of energy will be spent for receiving. So, when a node receives packets without transmitting any packet, it still consumes its battery for receiving the packet. When the total numbers of packets are increased (data and control packets), the energy consumed is also increased as shown in Fig. 12. We can infer that node 1 is unstable (when the node is dying out because of low battery capacity we call it as an unstable node, as it will not perform anymore like it performed when it was stable) when more than 5000 packets are involved in the network routing. So, the network has to be limited within 2000 packets without exceeding even a single byte. A network which holds unstable nodes with higher quantity of packets is undesirable. This again stresses the importance of a method, to track the optimum threshold level and once the level is reached, the node has to be made to enter the recovery phase which is discussed below. This is what MIST NTP (self recovery algorithm) is meant for which can make the node stable almost always.

The next possible variation of the energy consumption can be caused by different mobility node speed as explained in Figure 13. The stability analysis of node 1 is analysed at various node speeds. The curve shows the increasing energy consumption (decreasing battery capacity) by node 1, when the node speed is increased. This increase in the consumption can be due to increase in randomness of the node due to increase in node speed in MANETs. So, when the randomness increases the node is forced to spend more battery to send the data packets as compared to the nodes sending the data under less randomness. The node becomes unstable at higher speeds. A good network ideally should not have constraints over the node speeds and the mobility models. This increases the need to keep track of the optimum threshold rate of the battery capacity and cause the node to enter recovery phase. This makes the node become stable under any possible node speed and mobility model, whether random or uniform. It is obvious that if a node dies out and if it does not participate in forwarding the packets, then there is a drop in the overall number of packets reaching the destination node. The impact of unstable nodes which are dying out is shown in Fig. 14. This bar graph shows the number of packets drop that the destination going to suffer, when a specific node has lost all its battery capacity and is not notified to enter into the self recovery mode. Around 1300 packets are lost in case of node 1. If node 1 behaves as an unstable node then of the 5606 packets which are routed from source node, the destination node will receive only 4269 packets suffering loss in data packets, which is not trivial in fact.

Hence to reduce the unstable nodes in the network, the 'dying out node tracking method' and the self recovery algorithm of MIST is used.

CONCLUSION

The proposed MIST algorithm is implemented over the NTP routing protocol. The MIST algorithm comprises of three algorithms that can defend, detect, prevent or eliminate multiple attacks like modification attack, impersonation attack, sleep deprivation attacks and TTL attack at the same time.

Firstly, the Malicious Node Detection and Elimination (MDE) algorithm can detect and eliminate the malicious node from the network thereby eliminating the modification attack. Secondly, the authentication algorithm that uses the well known Diffie Hellman key exchange with RSA algorithm can protect the network against impersonation attack. Thirdly, Sleep deprivation torture attack can be eliminated by the combined effort of self recovery and "dying out node" tracking algorithms. The battery capacity is one of the prime features that control the network. If the battery capacity is less, the node becomes unstable and eventually the node fails to route the packets due to blockage of that particular node. So, this raised the necessity of the continuous dying out node tracking method, at an optimum threshold remaining battery capacity value. This technique can predominantly help the nodes in conserving their battery life. Finally, the TTL attack is detected and eliminated by using the same beacon packet. The malicious node inducing the TTL attack can be traced by the MDE algorithm with the aspects of NTP routing algorithm. Thus the multiple attacks are eliminated simultaneously by our novel set of algorithms which combines to form MIST NTP.

REFERENCES

Bing, W., C. Jianmin, W. Jie and C. Mihaela, 2006. A Survey of Attacks and Countermeasures in Mobile Ad Hoc Networks. Springer Link Publications, Springer Link Press USA., pp: 1-38. DOI: 10.1007/978-0-387-33112-6-5

Bo, S.M., H. Xiao, A. Adereti, J.A. Malcolm and B. Christianson, 2007. A performance comparison of wireless ad hoc network routing protocols under security attack. Proceedings of the International Symposium on Information Assurance and Security, Aug. 29-31, IEEE Xplore Press, pp: 50-55. ISBN: 0-7695-2876-7, DOI: 10.1109/IAS.2007.35

Diffie, W. and M.E Hellman, 1976. New directions in cryptography. IEEE Trans. Inform. Theory, 22: 644-654. ISSN: 0018-9448, DOI: 10.1109/TIT.1976.1055638

Gerhards-Padilla, E., N. Aschenbruck, P. Martini, M. Jahnke and J. Tolle, 2007. Detecting black hole attack in tactical MANETs using topology graph. Proceedings of the 32nd IEEE Conference on Local Computer Networks, Oct. 15-18, IEEE Xplore Press, Dublin, pp: 1043-1052. ISBN: 0-7695-3000-1, DOI: 10.1109/LCN.2007.72

Hu, Y., A. Perrig and B. Johnson, 2002. A secure on demand routing protocol for ad hoc networks. Proceedings of the eighth annual conference on Mobile Computing and Networking (ACM MobiCom), Sept. 23-28, ACM Georgia, pp: 23-28. ISBN: 1-58113-589-0, DOI: 10.1.1132.4609

Hu, Y., A. Perrig and B. Johnson, 2003. Rushing attack and defence in wireless ad hoc networks routing protocols. Proceedings of the 2nd ACM Workshop on Wireless Security, Sept. 19-19, ACM New York, pp: 1-11. ISBN: 1-58113-769-9, DOI: 10.1145/941311.941317

Khan, D., P. Ball and G. Childs, 2008. Extending the lifetime of multi hop ad hoc networks by managing the use of relay nodes. Proceedings of the CNSDSP 6th International Symposium Communication Systems, Networks and Digital Signal Processing, July 23-25, IEEE Xplore Press, pp: 657-661. ISBN: 978-1-4244-1875-6, DOI: 10.1109/CSNDSP.2008.4610702

Khatri, P., M. Rajput, A. Shastri and K. Solanki, 2010. Performance study of ad-hoc reactive routing protocols. J. Comput. Sci., 6: 1159-1163. DOI: 10.3844/jcssp.2010.1159.1163

Kurosawa, S. and A. Jamalipour, 2007. Detecting blackhole attack on AODV-based mobile ad hoc networks by dynamic learning method. Int. J. Network Security, 5: 338-346. ISSN: 1816-3548

Radha, S. and S. Shanmugavel, 2003. Implementation of node transition probability based routing algorithm for MANET and performance analysis using different mobility models. J. Commun. Networks, 5: 204-214. ISSN: 1976-5541

Ren, W., Y. Kim, J.-Y. Jo, M. Yang and Y. Jiang, 2007. IdSRF: ID-based secure routing framework for wireless ad-hoc networks. Proceedings of the International Conference on Information Technology, Apr. 2-4, IEEE Xplore Press, pp: 102-110. ISBN: 0-7695-2776-0, DOI: 10.1109/ITNG.2007.104

Rivest, R., A. Shamir and L. Adleman, 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21: 120-126. DOI: 10.1145/359340.359342, http://people.csail.mit.edu/rivest/Rsapaper.pdf

Sanzgiri, K. and M. Belding-Royer, 2002. A secure routing protocol for ad hoc networks. Proceedings of the 10th IEEE International Conference on Network Protocol, Nov. 12-15, IEEE Xplore Press, pp: 1-10. ISBN: 0-7695-1856-7, DOI: 10.1109/ICNP.2002.1181388

Stajano, F. and R. Anderson, 1999. The resurrecting duckling: security issues for ad hoc wireless networks, security protocols. Springer-Verlag, Lecture Notes Comput. Sci., 1796: 1-11. DOI: 10.1007/10720107_24

Vaithiyanathan, S.R. Gracelin, E.N. Elizabeth and S. Radha, 2010a. A novel method for detection and elimination of modification attack and TTL attack in NTP based routing algorithm. Proceedings of the International Conference on Recent Trends in Information, Telecommunication and Computing, Mar. 12-13, IEEE Xplore Press, Kochi, Kerala, pp: 60-64. ISBN: 978-1-4244-5956-8, DOI: 10.1109/ITC.2010.23

Vaithiyanathan, S.R. Gracelin, E.N. Elizabeth and S. Radha, 2010b. A novel method for multiple attacks in NTP based routing algorithm. Proceedings of the International Conference on Wireless communication and Sensor computing, Jan. 2-4, IEEE Xplore Press, Chennai, pp: 1-6. ISBN: 978-1-4244-5136-4, DOI: 10.1109/ICWCSC.2010.5415901

Wander, A.S, N. Gura, H. Eberle, V. Gupta and S.C Shantz, 2005. Energy analysis of public-key cryptography for wireless sensor networks. Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom), Mar. 8-12, IEEE Xplore Press, pp 324-328. ISBN: 0-7695-2299-8, DOI: 10.1109/PERCOM.2005.18

Yi, P., Z. Dai and S. Zhang, 2005. Resisting flooding attack in ad hoc networks. Proceedings of the IEEE Conference on Information Technology: Coding and Computing, Apr. 4-6, pp: 657-662. ISBN: 0-7695-2315-3, DOI: 10.1109/ITCC.2005.248