A Systematic Approach to IP traceback using MANET's

A.Sri Satya Kalyani
M.Tech
MVGRCE, Vizianagaram

Mrs.P.Parimala
Assistant Professor
MVGRCE, Vizianagaram

Dr.P.Satheesh
Associate Professor
MVGRCE, Vizianagaram

Abstract:

The previous system describes about a hybrid IP Trace back scheme in wired Mesh Network with efficient packet logging and marking. Aim of Packet logging is to have a fixed storage requirement for each router to maintain the identification information of packet payload. In packet logging there is no refreshment for tracking routers information dynamically. In this paper we proposed an IP Trace backing System on MANET in which the packets are routed dynamically on wireless AdHoc networks using the packet logging and Marking. Packet marking reduces the retransmission or duplication in transmission. Normally MANET allows low cost and time to delete and modify the nodes. As a first step a MANET is generated dynamically. Selected transmission will be fully controlled by router to mark the packets to change the path of transmission for maintaining the log information. The MANET will be fully controlled by selected router in all aspect of communication to overcome the spoofing DDOS and intrusion. The theoretical analysis and experiment results illustrate the working of Trace Backing scheme in Wireless AdHocnetworks. ***(Index terms: MANET, DDOS, marking, logging)***

1 Introduction:

IP traceback is a name agreed to any scheme for the formative derivation of a packet on the Internet. The source address in an IP packet can be spoofed allowing for DOS attacks since the authentication of source IP address is not guaranteed. The difficulty of finding the source of a packet is called the IP traceback problem. IP Traceback is a critical ability for identifying sources of attacks and instituting protection measures for the Internet. Most existing approaches to this problem have been tailored toward DoS attack detection. Such solutions want high numbers of packets to converge on the attack path(s).

Like MRT and MORE, RIHT marks interface numbers of routers on packets so as to trace the path of packets. Since the marking field on each packet is limited. Our packet marking scheme may need to log the marking field into hash table and store the table index on the packet.we repeat this marking/logging process untilthe packet reaches its destination. After that, we can reverse such process to trace back to the origin of attack packets.

MANET is a wireless communicated network. Once the network is established, the routes are monitored continuously by aggregated flows. In this approach, we propose a new framework on MANET(Mobile Ad Hoc Network).Here, in MANET the flows are distributed as Single Cast, Multicast & Broadcast. The Distributed Control plan is applied to each secure multipath aggregation through the MANET. Centralized forwarding from Root node to all nodes which was in MANET will be secured by the connectivity. The experiment results evolves the simulation results, are included to validate the performance of this framework.

The mobile AD Hoc Network framing is created. Many nodes have to be generated by levels. The 3 ways of data flow MANET's are Single cast, Multi Cast, Broad Cast. In Single cast, the flow will be from one node to particular nodes. In Multicast, the flow will be from multiple nodes to certain nodes. In Broad Cast, the flow will be from Root Node to any node in Manet.

Due to the complex nature attacks in hybrid networks, in the existing system we introduce a new approach that dynamically route the packets by reconstructing, while getting Intrusions, is implemented in MANET. Because the wireless communication is completely unsecure rather than wired networks. This is a challenging task to resolve the problem when attacks and conflicts are occurring in wireless.

a. Network Model:

Create a Wireless Mobile AdHoc network which is also known as MANET that is having the number of nodes over the network without physical communication. The trusted nodes, Routers should be present for monitoring the respective region of nodes. The MANET contains main router as a root for all these region nodes of wireless network. The routers in each level are also called as trusted nodes. These trusted nodes will act as secure nodes to permit the transactions between different levels of nodes. The root router manages all region of router nodes information within their logs for tracing the paths in network.

MANET generation

Terminology:

$\sum_0^{n-1} N \rightarrow Total\ no\ of\ nodes$

MANET

$\sum_0^{n-1} L \rightarrow Total\ no\ of\ Levels$

$S_c \leftarrow$ single casting

$M_c \leftarrow$ Multi casting

$B_c \leftarrow$ Broad casting

$S_N (s_c) \rightarrow$ Source nodes for Single casting

{

$\sum_0^{n-1} D_{sc} \leftarrow$ $destination\ nodes\ for$ Single casting

$\sum_0^{n-1} S_N\ (M_C) \leftarrow$ Source nodes for multi casting

$\sum_0^{n-1} D_{NS}\,(M_C) \leftarrow destination\ nodes\ for\ multi\ casting$

$A_N \leftarrow$ Admin node

MANET framing:→ "FRAME"

```
        Input ← no of peers
        Output ← N, $\lambda$=2, $A_N \leftarrow$ 0;
    Count=0; Level=0
        Loop : for each node m is n
        $A_n \sum N \leftarrow m$
        Count ++
        L=0 →n
        If l={0,2,4,8,16}
        Establish (L)
        $A_n E$ L
        End loop
        $\sum_0^{n-1} N \leftarrow A_n$
```

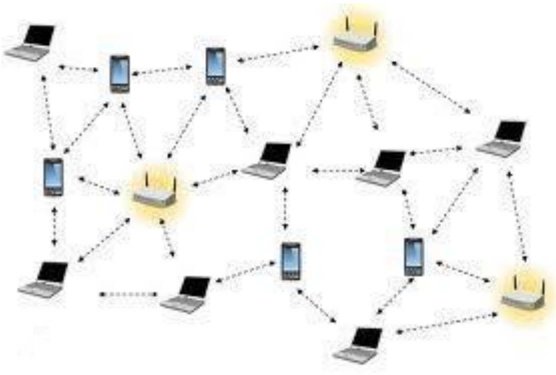


Fig 1. Mobile AdHoc Network

We introduce a IP traceback system which makes routers' interface numbers and integrates packet logging with a hash table to deal with logging and marking issues in IP traceback. It is designed to reach the following properties:

1) The logged information which is maintained with the hash table of paths over network is dynamically updated when a node is released or added.

2) Our scheme achieves zero false positive and false negative rates in attack-path reconstruction.

3) Getting best path reconstruction dynamically in MANET while encountering attacks.

b. Traceback Scheme in MANET:

The way to transmit packets over the wireless network from node to node is a big issue without attacks. In this paper, we propose a best and dynamic solution to resolve the problem. The MANET network is presently having less cost and is a dynamic network which maintains the wireless transmission throughout the network. For that we are implementing a technique which will manage the routers by updating paths and maintain the hash tables all by it.

Packet logging is implemented in each router for considering the paths while packet is transmitted. The Log information predicts the whole calculation of whether the packet is genuine or corrupted.Packet marking is for marking each node and router in which the packet is travelling through. Initially a packet is marked with 0. And finally contains the information about its path direction through which route it has travelled. Hash table maintains the list of paths. Duplicated paths are not allowed in this table and an unauthorized node cannot be considered here.

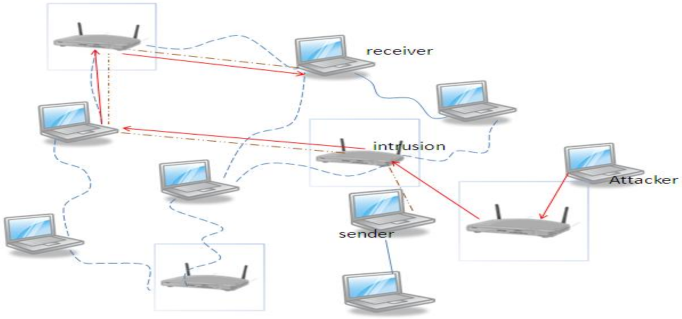


Fig 2. Data Transfer example figure

Hash Table:

Source node	Destination node	Available paths	Rate of Transmission
192.168.0.102	192.168.0.100		100mbps
. .	. .	. .	80mbps

Packet Design:

offset	Protocol	Source address	Destination Address	Travelling path	Payload information
0	TCP	192.168.0.102	192.168.0.100	ROUTE	AVERAGE

c. Packet marking and Logging:

Marking interface numbers of routers on packets so as to trace the path of packets. Since the marking field on each packet is limited, our packet-marking scheme may need to log the marking field into a hash table and store the table index on the packet. We replicate this marking/logging process until the packet reaches its destination. After that, we can invalidate such process to trace back to the source of attack packets.

1) A router creates an interface table and numbers of the upstream interfaces from 0 to Ri-1 in advance.

2) A router knows whether a packet comes from a router or a local network.

3) Such a traceback scheme is viable on every router.

4) The traffic route and network topology may be changed, but not often.

d. Dynamic Path Selection:

This anonymous protocol seeks to achieve anonymity with the minimal use of encryption and nullify the requirement of padding of data packet to prevent traffic analysis. In the protocol the next hop is dynamically selected by the router. This makes traffic analysis for a malicious router difficult as the traffic flow is erratic and confuses the adversary.

2. RELATED WORK

Papadimitriou and Haas proposed a secure routing protocol for MANETs using a security association between source and destination to validate the integrity of a discovered route. Sanzgiri has proposed cryptographic ways to secure routing in MANETs where in every intermediate node verifies the integrity of the message and then forwards it to the next node. Certificates are used by source and destination nodes to get the public key of each other. ASR uses anonymous virtual circuit in routing and data forwarding where each node does not know its immediate upstream nodes and immediate downstream nodes. Using a special anonymous signaling procedure, the node only knows the physical presence of neighboring Ad-hoc nodes. The session key of the route between every pair of the intermediate nodes is determined when a node forwards reply packet to its upstream nodes. Although the above mentioned anonymous routing techniques can provide a certain level of anonymity, an external adversary can still monitor the transmitted packets to identify the communication peers.

3. System Model

We explain here the notations, assumptions and the system model. Every node in the network maintains ART and ARC. Destination maintains PIT and IRT as well. Source node starts with the route discovery message (route Request) by flooding it to all neighbors. Requestid is embedded in it.

Notations used:

S : Sender R : Receiver

M : Message D : Data

X : Intermediate node E : Encrypt function

Cc Count: Criss-cross count ccTimer: Criss-cross timer

PUN : Public key of N PRR : Private Key of N

D : Decrypt function ccTable : Criss cross Timer Table

ART: Anonymous Routing Table

IRT: Intermediate Routing Table<pathID, path_of_message>

PIT :Path Info Table <pathID, nodeID, nextHop>

ı :Set inclusion, modeled as appending at the end of set (array)

ARC :Anonymous Routing Cache <reqID, ccCount>

exists(x,z) :returns true if table z has record mapped to x, otherwise false.

getCnt(x) : returns ccCount value from the record <x, ccCount> of ARC, if no such record

found then return false.

setCnt(x, y) : sets ccCount value from the record <x, ccCount> of ARC to y.

expired(x) :returns true if timer mapped to x is expired else false.

Example of Secure Anonymous Routing Protocol:-

Data portion D of the message contains pathID, Source(S), Destination (D) and Nonce's which is encrypted using the public key of destination (PUD). Message also contains I, set of all nodes traversed by routeRequest message to reach destination and PUD. Every entry in I is encrypted using PUD. ccCnt indicates the number of more routeRequest messages with same requested that can be flooded by the same node. Initially assigned value to ccCnt is a parameter set by network administrator, subject to tuning. Upon receiving routeRequest message with given reqID first time, node makes an entry in ARC, inserting reqID and ccCnt. For subsequent receipts of routeRequest message with same reqID, node checks whether value of ccCnt is zero or not. If zero then ccCnt limit is reached and packet is discarded there itself. If not zero then ccCnt is decremented by 1 and message is forwarded to neighbors by appending its id (encrypted with PUD) in I field of the message.

For IRT entry<P1, < 0, 1, 4, 6>>, it updates PIT entries as <P1, 0, <1>>, <P1, 1, <4>>, <P1, 4,<6>>, <P1, 6, <>>. For <P1, < 0, 3, 4, 6>>, it updates PIT as <P1, 0, <1, 3>>, <P1, 3,<4>>, <P1, 4, <6>>, <P1, 6, <>>. This is used to construct routeReply messages, composed of routing table updates of en-route nodes. Destination node encrypts these PIT entries with the public keys of en-route nodes, in the sequence marked in the routeRequest message and onion routing [1, 6] is used to forward these updates to en-route nodes. N once R, F (Nonce's) are also added to message encrypted using public key of Source.

Here we assume that any node leaving the network does not cause the partition in the MANET. Every node(X) sends a signal to its neighbor, and updates the status of neighbors depending upon the reply. If any node NL discovers change in the topology of network then it searches <pathID, Z> in ART such that NL Z. If such entry is found then it sends the update message to all nodes in Z and removes NL from Z.

After removing the entry, if Z is empty then node floods the route invalidate message with corresponding pathID. Upon receiving the update message, node updates its ART. In case the node receiving the invalidate message is the one that started the communication with the corresponding pathID, it re-initiates route discovery.

4. Proposed Algorithm

4.1 Path Discovery Phase (Shuffling Approach)

```
Source initiates with routeRequest message<reqID, E(PUR,D), I> ;D=, <pathID, sourceID,
destinationID, NONCES>, I={E(PUR, S)} by sending to all neighbors.
X (≠R), an intermediate node receives routeRequest<reqID, E(PUR,D), I>message:
if ( exists(reqID, ARC) ı getCnt(reqID) ≠ 0)then
setCnt(reqID, getCnt(reqID) - 1) /* decrement the ccCount */
IIU{ E(PUR, X)} /* Append ID to the message*/
forward<reqID, E(PUR, D), I> to neighbors except the one from it received.
elif(exists(reqID, ARC) ı getCnt(reqID) = 0)then
Discard routeRequest message as ccCnt limit reached.
else
ARC ARCı {<reqID, ccCntUL>} /* Make entry in ARC */
IIU{ E(PUR, X)} /* append ID in message */
forward<reqID, E(PUR, D), I> to neighbors.
endif
Send acknowledgement to the node (sourceID) from which message is received.
R receives routeRequest message MRQ<reqID, sourceID, destinationID, E(PUR, D), I>:
Decrypt each entry of I private key, store decrypted values in I.
if ( exists(reqID, ARC) ı expired(ccTimerreqID) )then
Discard routeRequest message. /* Timer Expired */
elif ( exists(reqID, ARC) ı ¬expired(ccTimerreqID) ı getCnt(reqID) = 0))then
```

4.2 Construction of Routing Table entries for intermediate nodes

4.3 Updating ART of intermediate nodes

```
Discard routeRequest message. /* Criss-cross count limit reached */
elif ( exists(reqID, ARC) ı ¬expired(ccTimerreqID) ı getCnt(reqID) ≠ 0))then
setCnt ( reqID, getCnt(reqID) - 1)
pathID D(PRR, E(PUR, D))
IRT IRTU {<pathID, I ∪ {R} >}
else (¬exists(reqID, ARC), ARC) /* No entry found in ARC for reqID */
ccTableccTableU {<pathID, ccTimerMrq>} /* Set ccTimer*/
ARC ARCU {<reqID, 5>}
pathID D(PRR, E(PUR, D)
IRT IRTU {<pathID, I ∪ {R} >}
endif
/*Process entries in IRT with reqID for with ccCnt is zero or ccTimer is expired*/
for each <reqID, I> in IRT do
for each xi in I;xi≠R,do
if exists(<pathID, xi, Z>, PIT)then
Update ZZU{ xi+1} in PIT
else
PIT PITU {<pathID,xi, { xi+1}> }
endif
end for
end for
Constructing and sending Reply Message:
for each <pathID, I>in IRT do
I'= ı /* initialize I' as Null*/
for each xi in I, i=n…1 do /* Reverse the path for reply message */
I'=I' ∪ {xi}
```

```
endfor
temp=< NONCER , F(NONCES)>
for each xi in I', xi≠R do
Search <pathID, xi , Z> in PIT
if i=1 then /* for source node's case */
msg = msg + <S , E(PUxi, <<pathID, xi , Z>, temp>) >
elsemsg = msg + <xi-1 , E(PUxi, <pathID, xi , Z>) >
endif
end for
Send routeReply message MRP <R, xn-1, msg> to xn-1 /*<source, to, msg_data> */
end for
```

```
X receives the routeReply message MRP <Y, X, msg>:
/* extract the routing info sent by the destination and update ART */
<<pathID, X , Z >, nextHop, E(PUnextHop , msg) > = D(PRx, msg)
ART = ART ∪ {<pathID, Z>} /*Update routing table*/
if X=S then
<pathID, NONCER , F(NONCES)> = D(PRS, msg)
Send F(NONCER) to the destination.
else
forwardMRP<X, nextHop, msg>
endif
```

4.2 Data Communication Phase

Source-destination pair exchanges session key for regular data transfer. Source sends message with pathID prepended to the message. Every intermediate node will choosethe next hop dynamically from its ART corresponding to the pathID in the message. We have employed acknowledgement mechanism for detection of passive nodes.

5. Simulation Results

We have written our simulator using C in UNIX. All cryptographic operations are performed using OpenSSL Crypto API. MANET is constructed using 50 nodes, initially uniformly distributed. Source destination pairs are chosen randomly. Mobility of nodes is random, with constant speed. Once node becomes immobile, it waits there for fixed time. Maximum number of communicating pairs in MANET at a given time is assumed to be 20, chosen randomly. We use cc_cnt, and cc_timer metrics as global tunable parameters which are set by network administrator. By increasing the value of ccCnt, the number of paths discovered is more. However few of these paths might be longer ones. So the delay incurred on an average to reach the destination also increases.

6. Simulation Analysis

6.1 Anonymity Analysis

Identity Privacy: In our protocol, the identities of source and destination are known only to the two communicating parties, as we are using them only in the route request message and with encryption, there by not revealing them to intermediate nodes.Hence identity privacy is ensured.Route Anonymity: In our protocol, no adversary can trace a flow of packet because of random selection of next hop and there by leading to dynamic path selection. Any adversary on the route has no information about the path other than the next hop. As we have employed fixed size padding, we can introduce several dummy packets and reshuffle the actual packets in the buffer to eliminate the possibility of temporal analysis as defined.

6.2 Possible attacks

Route Rediscovery Attack: One possible attack is that adversaries send fake route update or invalidate packets to fool the intermediate nodes or source to begin route rediscovery process. In our protocol, only the nodes whose routing table has entry for the node leaving the network, can send the route invalidate, router discovery or route update messages which ever applicable as explained in the algorithm. So our proposed protocol is less vulnerable to the route rediscovery attack.Selfish Nodes or Byzantine nodes: Byzantine nodes can intercept packets, create routing loops, selectively drop packets, or purposefully delay packets. Our protocol uses the acknowledgement mechanism. If any node is dropping the packet thenacknowledgement will not be sent to sender. Even in presence of live communication link, if node is dropping packets then it can be detected as selfish node. And as we are choosing next hope dynamically at any intermediate node for routing, we can excludethis selfish node from the ART.

6.3 Cryptographic Overhead

In our protocol, we use cryptosystem of the shape onion just for path discovery. For electronic communication, information is encrypted by source with the destinations public key, i.e. end to end encryption; onion routing is not used here. So there's not abundant cryptanalytic over head concerned for traditional electronic communication section that leads to computational advantage.

Packet Routing

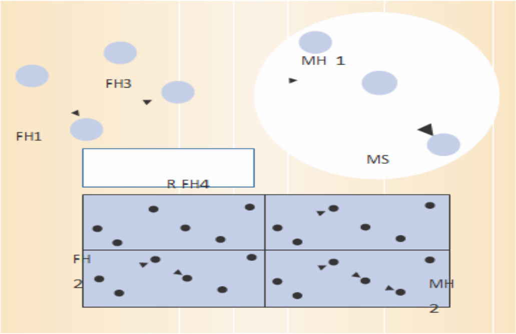


Fig 3. Packet Routing

Fixed Host

MH Mobile host

MSR Mobility support router

ON-DEMAND ROUTING ALGORITHMS:

Rather than looking forward to periodical broadcasts of obtainable routes, algorithms like Dynamic supply Routing (DSR) and Adhoc On-Demand Distance Vector Routing (AODVR) discover routes pro re data. as a result of the route to each mobile node isn't acknowledged at any given time, these algorithms should build and maintain routes. for instance, robots or autonomous sensors deployed in a locality inaccessible to humans may use easy painter routing protocols to transmit information to an impact center.

ROUTING MANETS:

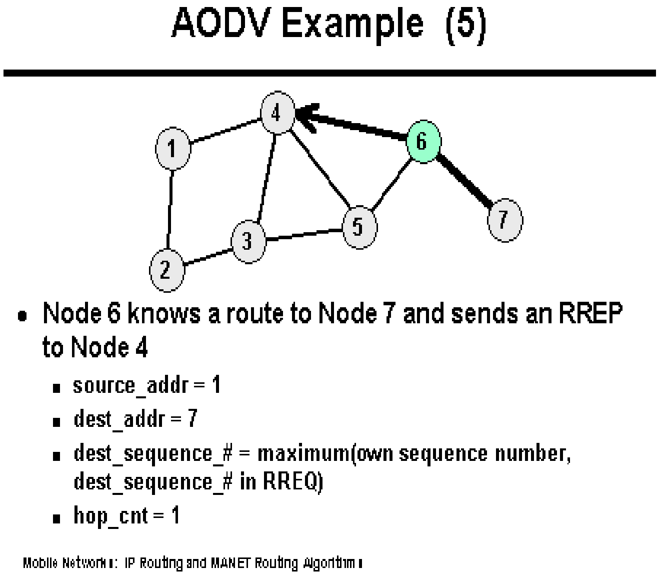
Efficient routing of packets is a primary MANET challenge. Conventional networks typically rely on distance-vector or link-state algorithms, which depend on periodic broadcast advertisements of all routers to keep routing tables up-to-date. In some cases, MANETs also use these algorithms, which ensure that the route to every host is always known. However, this approach presents several problems:

- periodically updating the network topology increases bandwidth overhead;
- repeatedly awakening hosts to receive and send information quickly exhausts batteries;
- the propagation of routing information, which depends on the number of existing hosts, causes overloading, thereby reducing scalability;
- redundant routes accumulate needlessly; and
- communication systems often cannot respond to dynamic changes in the network topology quickly enough.

MANETs use multi-hop rather than single-hop routing to deliver packets to their destination.

Dynamic Source Routing:

DSR is a fairly simple algorithm based on the concept of *source routing*, in which a sending node must provide the sequence of all nodes through which a packet will travel. Each node maintains its own *routecache, essentially* a routing table, of these addresses. Source nodes determine routes dynamically and only as needed; there are no periodic road casts from routers. The RREQ propagates through the network, collecting the addresses of all nodes visited, until it reaches the destination node or an intermediate node with a valid route to the destination node. This node in turn initiates the route reply process by sending a special route reply (RREP) packet to the originating node announcing the newly discovered route.



AODV Example (5)

- Node 6 knows a route to Node 7 and sends an RREP to Node 4
  - source_addr = 1
  - dest_addr = 7
  - dest_sequence_# = maximum(own sequence number, dest_sequence_# in RREQ)
  - hop_cnt = 1

Mobile Networks: IP Routing and MANET Routing Algorithms

Fig 4. Adhoc on-Demand Distance Vector Routing

With AODVR, a source node that wants to send a message to a destination for which it does not have a route broadcasts an RREQ packet across the network. All nodes receiving this packet update their information for the source node. Thus, unlike DSR, this approach does not use route caching. Instead, each node maintains only the next hop's address in a routing table, and these routing tables are updated all the way along the RREQ propagation path. The RREQ contains the source node's address, broadcast ID, and current sequence number as well as the destination node's most recent sequence number. Nodes use these sequence numbers to detect active routes. A node that receives an RREQ can send an RREP if it either is the destination or has a route to the destination with a corresponding sequence number greater than or equal to the sequence number the RREQ contains .In the latter case, the node returns an RREP to the source with an updated sequence number for that destination; otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ source address and broadcast ID, discarding any RREQ they have already processed. As the RREP propagates back to the source, nodes setup entries to the destination in their routing tables. The route is established once the source node receives the RREP. This algorithm also includes route maintenance facilities. For every route in a routing table, a host maintains a list of neighboring nodes using that route and informs them about potential link breakages with RERR messages. Each node also records individual routing table entries and deletes those not used recently.

AODVR offers several key advantages compared to DSR:

- It supports multicast by constructing trees connecting all the multicast members along with the required nodes
- Smaller control and message packets result in less network bandwidth overhead; and
- The need for only two addresses when routing—destination and next hop—rather than the entire sequence ensures good scalability because packet size does not depend on network diameter.

However, AODVR only works with symmetric links, and because it does not allow for multipath routing, new routes must be discovered when a link breaks down.

LINK-STATEROUTINGALGORITHMS

Link-state routing algorithms exploit the periodic exchange of control messages between routers, ensuring that the route to every host is always known and immediately providing required routes as needed. However, this pro activity comes at the cost of high bandwidth overhead. Adhoc link-state routing algorithms attempt to conserve bandwidth by reducing the size and number of control messages.

Optimized link-state routing

Classic link-state algorithms declare all links with neighboring nodes and flood the entire network with routing messages. Optimized link-state routing compacts control packet size by declaring only *multipoint relay selectors* ,a subset of neighboring links .To further reduce traffic ,OLSR uses only the selected nodes, called *multi point relays*(MPRs),to flood the network with routing messages. Each no deselects a set of neighboring nodes as MPRs, and these nodes rebroadcast packets received from the originating node. Each node maintains a table of MPR selectors and rebroadcasts every message coming from those selectors. In this way, the network distributes only partial link-state information, which OLSR can use to calculate an optimal route in terms of number of hops.

Each node periodically broadcasts hello messages containing information about its neighbors and a link status. Nodes select the minimal subset of MPRs among one hop neighbors to cover all nodes two hops away. Thus, every node in the two hop neighborhood must have a symmetric link to a given node's MPR set.

Topology based on reverse path forwarding

TBRPF broadcasts link-state updates via source trees that provide paths to all reachable nodes. It computes these source trees with partial topology information using a modification of Dijkstra's algorithm. Similar to OLSR, each node declares only part of its TBRPF uses both periodic broadcasts and differential updates to report updates, but each node can declare a full tree, leading to mation, violating network confidentiality.

Hacker can directly attack the network to remove away the messages, inject malicious messages or impersonate a node, and non repudiation. Compromised nodes also can launch attacks from within a network. On-demand and link-state routing algorithms do not specify a scheme to protect data or sensitive could lead to significant vulnerability in MANETs ,a security solution must be based on the principle of distributed trust the entire topology's link-state behavior. Each route update travels along a single path to every node on a source tree; leaves do not forward updates .Nodes discover a neighbor using differential hello messages that only report changes in the neighborhood, which makes the messages smaller than those in OLSR. This algorithm is useful in dense mobile networks.

HYBRID APPROACH

A recently proposed hybrid approach captures the advantages of on-demand and optimized link- state routing for wireless sensor networks. This algorithm discovers the route to each node only when it is required. However, route exploration does not occur through simple flooding but through a mechanism similar to multi point relays.

The algorithm defines three types of nodes: master, gateway, and plain. A group of nodes elects a master to form a piconet and then synchronizes and maintains the acquaintance list. A node can be in a root position in only one piconet, but it can be a plain member in any number of piconets. Portal nodes belong to two or more piconets. Only masters and gateways forward routing information; plain nodes receive and process this information, but they do not forward it. Simulation shows that this algorithm works best when the pioneers are densely populated; other-wise, it degrades to simple network flooding. Future research should focus on using some well-defined and accepted metrics, such as power consumption, to compare various adHoc routing approaches.

SECURITY MANETS

The use of wireless links makes MANETs susceptible to attack .Eaves droppers can access confidential information, violating network confidentiality. Hackers can directly capture the network to remove away the messages, inject malicious messages, or impersonate a node, which does not obey the status of availability, integrity, authentication, and non repudiation. Compromised nodes also can launch attacks from within a network. On-demand and link-state routing algorithms do not specify a scheme to protect data or sensitive routing information. Because any centralized entity could lead to significant vulnerability in MANETs, a security solution must be based on the principle of distributed trust.

This is similar to the dilemma posed by the classic Byzantine generals problem, in which a general commands each division of the army, and some of the generals, who communicate via messenger, are traitors. All loyal generals must decide upon the same plan o faction—that is, a small number of traitors cannot cause the loyal generals to adopt a bad plan. The same holds for MANETs: A number of compromised nodes cannot cause the network to fail. Although no single node in a MANET is trustworthy, threshold cryptography can distribute trust to an aggregation of nodes. This scheme lets *n* parties share the ability to perform a crypto- graphic operation such that any *t* parties can do it together, while up to *t*−1 parties cannot perform the operation. However, dividing a private key into *n* shares and constructing *t* partial signatures is nontrivial given that traditional key distribution schemes either do not apply to the ad Hoc scenario or are not efficient for resource constrained devices. Combining identity based techniques with threshold cryptography can achieve flexible and efficient key distribution. After distribution, a combiner can verify the *t* signatures and compute the final signature for the certificate. In this way, up to *t*−1 compromised nodes cannot generate a valid certificate by themselves.

If a large number of nodes are compromised, attributing fault to a specific malicious node is impossible. A proposed algorithm addresses this problem by limiting the possible fault location to the link between two adjacent nodes; as long as a fault-free path exists between two nodes, they can establish a secure communication link even if most nodes in the network are compromised. In addition, this algorithm can detect selfish nodes that refuse to cooperate with other nodes. If their behavior is the result of a denial-of-service attack rather than power-savings activity, the algorithm can isolate the selfish nodes. Wireless research today primarily focuses on the functional aspect of MANETs—improving the delivery of packets from one node to another. However, as technology matures, non- functional properties such as semantics and security will play the leading role. The challenge lies in managing these two layers, which are orthogonal to each other. If ADHOC communication is to be the foundation for pervasive computing, we must be able to seamlessly inter connect different platforms and devices, offer services on demand, and make it all secure and trusted.

CONCLUSION:

Once the network is created, the routes are controlled continuously by aggregated flows. In this approach, we introduce a new framework on MANET(Mobile Ad Hoc Network).Here, in MANET the flows are shared as Single Cast, Multicast & Broadcast. The Distributed Control plan is implemented to each protected multipath aggregation via the MANET.

We introduced an IP Trace backing System on MANET where the packets are directed randomly on wireless AdHoc networks using the packet logging. Packet marking decreases the duplication in sending the packets. MANET allows low cost and time to remove and change the hops. Initially a MANET is created automatically. Selected transmission will be fully under router to mark the packets to alter the path of transmission for keeping the log information. The MANET will be under selected router in all aspect of communication to overcome the spoofing DDOS and attack. The theoretical analysis and experiment results explain the working of Trace Backing scheme in Wireless AdHocnetworks.

REFERENCES:

1. D.B. Johnson, D.A. Maltz, and Y-C.Hu,“ The Dynamic Source Routing Protocol for Mobile AdHoc Networks (DSR),”IETF Mobile AdHoc Networks Working Group, Internet Draft, work in progress,15th Apr,2003.

2.C.E.Perkins,E.M.BeldingRoyer,andS.R.Das,“Ad Hoc On Demand Distance Vector (AODV) Routing,” IETF Mobile AdHoc Networks Working Group, Internet Draft,workinprogress,17Feb.2003.

3. S. R. Das, C.E. Perkins ,and E.M. Belding-Royer, “Performance Comparison of Two On-Demand Routing Protocols for AdHoc Networks, ”*Proc .IEEE Infocom2000,*vol.1,IEEEPress,2000,pp.3-12.

4. P. Jacquetet al. ,“Optimized Link State Routing Protocol for Ad Hoc Networks, ”*Proc .IEE EInt’ l Multi Topic Conf., 2001*,IEEEPress,2001,pp.62-68.

5. R. Ogier, F. Templin ,and M. Lewis,“ Topology Dissemination Based on Reverse-Path Forwarding (TBRPF),”IETF Mobile AdHoc Networks Working Group, Internet Draft, work in progress, 14Oct.2003.

6. N. Milanovicetal.,“ Bluetooth AdHoc Sensor Network,”*Proc.2002 Int’l Conf. Advances in Infrastructure fo re-Business, e-Education ,e-Science, and e-Medicine on th eInternet,* Scuola Superiore G.Reiss Romoli, 2002; www.informatik.hu-berlin.de/

~milanovi/bt_adhoc_sensor.pdf.

7. I.Stojmenovic and X.Lin,“Power-Aware Localized Routing in Wireless Networks,” *IEEE Trans.Parallel and Distributed Systems,* vol.12,no.11,2001, pp.1122-1133.

8. L. Lamport ,R.E. Shostak, and M. Pease, “The Byzantine Generals Problem, ”*ACM Trans. Programming Languages and Systems,* vol.4,no. 3,1982,pp.382-401.

9. Y. Desmedt, “Some Recent Research Aspects of Threshold Cryptography,”*Proc.1stAnn.Workshop InformationSecurity,*LNCS1396, Springer-Verlag,1997,pp.158-173.

10. A.Khalili, J. Katz, and W.A. Arbaugh, “Toward Secure Key Distributionin Truly AdHoc Networks,”*2003 Symp. Applications and the Internet Workshops(SAINT03Workshops),*IEEECSPress,2003, pp.342-346.

11. B. Awer buchetal.,“ An On-Demand Secure Routing Protocol Resilentto Byzantine Failures,” *Proc. ACM Workshop Wireless Security,* ACM Press, 2002,pp.21-30.