A Study on Data Security in MANETS

Tummala Sasanth , Dr Syed Umar ,Divya Bellam

Department of ECM,
KL University, A.P. INDIA.

Abstract— **A mobile ad hoc network (MANET) is a self-configuring infrastructure-less network of mobile devices connected by wireless. *Ad hoc* is Latin and means "for this purpose". Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The primary challenge in building a MANET is equipping each device to continuously maintain the information required to properly route traffic. Such networks may operate by themselves or may be connected to the larger Internet. MANETs are a kind of Wireless ad hoc network that usually has a routable networking environment on top of a Link Layer ad hoc network. Due to their applications in situations such as emergencies, crisis management, military and healthcare, message security is of paramount importance in mobile ad-hoc networks. However, because of the absence of a fixed infrastructure with designated centralized access points, implementation of hard-cryptographic. Security is a challenging prospect. In an adverse environment, both route discovery and data transmission are vulnerable to a variety of attacks. The trust based multi-path routing ensures secure discovery of multiple path between source and destination. Self-encrypted parts of a message are transmitted through these paths. Therefore it is difficult for malicious nodes to gain access to the minimum information required to break through the encryption strategy. Results show that our method is much more secure than other existing trust based multipath routing protocols.**

Keywords—**Wireless sensor networks, adaptive topology, topology control, manets, security**

I. Introduction

MANETS do not rely on extraneous hardware, which makes them an ideal candidate for rescue and emergency operations. They are built, operated and maintained by their constituent wireless nodes. These nodes generally have a limited transmission range, so each node seeks the assistance of its neighboring nodes in forwarding packets. In order to establish routes between nodes which are farther than a single hop, specially configured routing protocols are engaged. The unique feature of these protocols is their ability to trace routes in spite of a dynamic topology. Communication in mobile ad hoc networks comprises two phases, route discovery and data transmission. In an adverse environment, both phases are vulnerable to a variety of attacks. First, misbehaving nodes can disrupt the route discovery by impersonating the destination, by responding with stale or corrupted routing information, or by disseminating forged control traffic. This way, attackers can obstruct the propagation of legitimate route control traffic and adversely influence the topological knowledge of benign nodes. However, misbehaving nodes can also disrupt the data transmission phase and, thus, incur significant data loss by tampering with, fraudulently redirecting, or even dropping data traffic, or injecting forged data packets.

To provide complete security in both phases of a MANET, we require secure routing protocols, since nodes involved in the routing cannot by themselves ensure the secure and undisrupted delivery of transmitted data. This is so, since misbehaving nodes could abide with the route discovery and be placed on utilized routes. But then, they could tamper with the in-transit data in an arbitrary manner and degrade network operation. Upper layer mechanisms, such as reliable transport protocols, or mechanisms currently assumed by the MANET routing protocols, such as reliable data link or acknowledged routing, cannot cope with malicious disruptions of data transmission. In fact, the communicating nodes may be easily deceived for relatively long periods of time, thinking that the data flow is undisrupted, while no actual communication takes place. One way to counter security attacks would be to cryptographically protect and authenticate all control and data traffic. But to accomplish this, nodes would have to establish the necessary trust relationships with each and every peer they are transiently associated with, including nodes that just forward their data.

Even if this was feasible, such cryptographic protection cannot be effective against black hole and grey-hole attacks, with misbehaving nodes simply discarding data packets. In this paper we propose a novel method through which we are able to provide security in both phases. To enhance security in the routing phase, a trust based multipath routing protocol is used. It discovers a secure, trustworthy path from source to destination with minimal overhead. Multiple node disjoint paths are discovered to enhance the security of the data delivery phase. Misbehaving nodes are detected and exempted from such paths using the trust value of the nodes. Sending confidential data on one path helps attackers to get the whole data easily, whereas sending it in parts on different disjointed paths increases the confidentiality robustness, because it is almost impossible to obtain all the parts of a message fragmented and sent on multiple paths existing between the source and the destination.

The rest of this paper is organized as follows. In Section 2 the related work is given, followed by a detailed description of our method in Section 3. In Section 4 we evaluate the efficiency of our method through exhaustive simulation. An analysis of the proposed method is also presented. Finally, the last section concludes the paper.

2 Related Work

This section surveys and analyzes existing methods to enhance security in hostile and dynamic MANET environments. Some of these methods aim to enhance security in the routing phase, while others concentrate on data security. A trust-based routing is proposed by Pirzada [10] in which the trust agent derives trust levels from events that are directly experienced by a node. A reputation agent shares trust information about nodes with other nodes in the network. A combiner computes the final trust in a node based upon the information it receives from the Trust and Reputation agents. Trust is computed using direct and indirect information. The trust value is propagated by piggy backing the direct trust value of the nodes along with RREQ packets [11]. Each time a packet is sent or forwarded, the forwarding node scans the routing tables for all alternate paths leading to the destination. It compares the direct trust value of all next hops in this path and selects the one with the highest trust value. However, The network overhead is increased because of the indirect information used in trust calculation, as it uses more control packets for advertising trust, calculating observed trust and issuing certificates in the trust calculation. Wang *et al.* [15] have also proposed a Routing Algorithm based on trust. They have assumed that the trust values of all nodes are stored at each node in advance. Trust for a route is calculated at the source node based on the weight and trust values are assigned to the nodes involved in the path at the source node. Weights are assigned by the source node ranging from 0 to 1. The protocol uses the path with the largest trust value of route and minimum hop count from among multiple route options, as metrics, unlike the standard DSR protocol that only uses minimum hop count. However, they have used a forward trust model to find the path from source to destination. So trust is embedded only in the RREQ packet when it is forwarded. Each node evaluates only its previous node and the source node evaluates all the nodes involved in path. But we believe that trust is asymmetric, so mutual trust information should be used. A trust-based multi path DSR protocol was proposed by us [12] which uses multi-path forwarding approach. In this approach each node forwards the RREQ if it is received from a different path. Through this method we are able to detect and avoid misbehaving nodes which were previously included due to vulnerability in DSR route discovery. In the traditional DSR protocol [5] when a node receives a RREQ packet, it checks if it has previously processed it; if so, it drops the packet.

A misbehaving node takes advantage of this vulnerability and forwards the RREQ fast, so that the RREQs from other nodes are dropped and the path discovered includes itself. In this protocol, each node broadcasts the packet by embedding trust information about the node from which the packet is received. At the source node a secure and efficient route to the destination is calculated as the weighted average of the number of nodes in the route and their trust values. In TDSR [16] model, trust among nodes is calculated as a combination of direct trust and indirect trust. The direct trust score is modified when misbehavior has occurred by a number of times exceeding a threshold. The indirect trust score is modified when a node receives a message reported by neighbor nodes. If the trust score of a node in the table has deteriorated so as to fall out of a tolerable range, such nodes are added to the blacklist. In the Route Discovery phase, when node *A* sends a RREQ packet to node *B*, *B* looks up its blacklist to find whether node *A* is in it. If not, it forwards the packet.

The trust-embedded AODV (T-AODV) routing protocol [12] was designed to secure an ad hoc network from independent malicious nodes by finding a secure end-to-end route. In this protocol, trust values are distributed to the nodes a priori. In the route discovery phase the RREQ packet header contains a *trust level* field, in addition to the other fields. Each intermediate node rebroadcasts the RREQ after modifying the *trust level* by including the trust level of the node that sends it the RREQ to. All the RREP are sent by the destination. The source node selects the route with the highest value of the *trust level* metric.

Narula *et al.* [8] proposed a novel method for message security using trust-based multi-path routing. The Pirzada model [10, 11] is used for assigning trust levels to the nodes in the network. The trust level is assigned in discrete values, from -1 to 4, which signify complete distrust to complete trust. The paths between the source and destination are found using DSR. The trust levels assigned to the nodes are used to define the maximum number of packets which can be routed via these nodes.

Nodes having lower trust values are given lesser number of encrypted parts of a message, making it difficult for malicious nodes to access the information in the message. A node with trust level 0 is not given any message and all the packets received from a node having trust level as -1 are dropped. A node with trust level 4 can read the message. Hence, only those nodes that are completely safe can read the message. The authors have used message encryption and decryption as proposed in [4]. However when the malicious nodes work in collaboration, they have high trust levels. Therefore such nodes are able to get higher number of self encrypted packets and are able to decrypt the message.

The Security Protocol for Reliable Data Delivery (SPREAD) scheme addresses data confidentiality and data availability in a hostile MANET environment [7]. The confidentiality and availability of messages exchanged between the source and destination nodes are statistically enhanced by the use of multipath routing. At the source, messages are split into multiple parts that are sent out via multiple independent paths. The destination node then combines the received parts to reconstruct the original message. SPREAD scheme assumes link encryption between neighboring nodes, with a different key used for each link. The threshold Secret Sharing algorithm [14] is used to divide messages into multiple parts. The SMT [9] scheme operates on an end-to-end basis, assuming a Security Association (SA) between the source and destination nodes, so no link encryption is needed. This SA between end-nodes is used to provide data integrity and origin authentication, but it could also be utilized to facilitate end-to-end message encryption. The scheme works on top of existing secure routing protocols, which cannot by themselves ensure data security. SMT provides an explicit end-to-end secure and robust feed-back mechanism that allows for fast reconfiguration of the path-set in case of node failure or compromise. Each path is continually given a reliability rating that is based on the number of successful and unsuccessful transmissions on that path. SMT uses these ratings in conjunction with a multipath routing algorithm to determine and maintain a maximally secure path-set and adjust its parameters to remain efficient and effective. SMR (Split Multi-path Routing) [6] is based on DSR [5] attempts to discover maximally node disjoint paths. The routes are discovered on demand in the same way as in DSR. From the received RREQs, the destination selects two multiple node disjoint paths and sends a Route RE-Ply (RREP) packet. However, no method to take care of misbehaving nodes has been implemented. Thus we find that there is no global solution to enhance security in both the phases of MANETs. Trust based approaches are able to detect and isolate misbehaving nodes in the routing phase. But, secure routing does not completely address the core problems in secure communication. For example, it cannot prevent misbehaving nodes on the communication path from eavesdropping or modifying data traffic. Similarly, secure routing cannot detect or prevent packet loss because a misbehaving node can abide well in the route discovery phase and be placed on utilized routes. Later, it could tamper with the in-transit data in an arbitrary manner and degrade the network performance.

3 Trust Establishment

We use a variation of the trust models used in [10] and [11] in our algorithm. A node is assigned a discreet trust level in the range of 0 to 1. A trust level of 1 defines a complete trust and a trust level of 0 defines a complete distrust.

3.1 Trust Level Assignment

The trust level assigned to a node is a combination of direct interaction with its neighbors and the recommendations from its peers. A node assigns a direct trust level to its neighbor on the basis of the acknowledgements received. If the neighbor sends a prompt acknowledgement of the packet received, it is assumed that the node is not involved in a resource intensive brute-force attack and hence is assigned a higher trust level. The direct trust is then combined with the trust recommendation from its peers and a final trust level is assigned to it. Note that these trust levels are assigned dynamically and are coached by a node for performance enhancement. The trust recommendations are piggy backed on DSR routing packets.

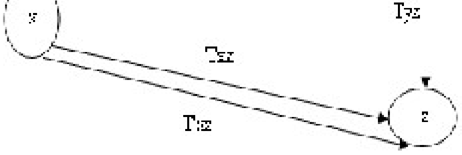

Figure 1: Trust assignment

Let us consider Figure 2. Let *Txy* represents the direct trust in node *Y* by node *X* and let *Tyz* represents the trust recommended by the node *Y* in node *Z*. If *Txz* represents the direct trust of node *Z* in node *X*, then the trust assigned by *X* in *Z* is given below [7]:

T0xz = 1 ¡ (1 ¡ *Txz*)(1 ¡ *Txyz*)*;* where

Txyz = 1 ¡ (1 ¡ *Txy*)*Tyz:*

The trust levels are normalized to integer values using standard methods. Each node is given an integer trust value lying between 0 and 1. If a new node joins the network, it sends a hello packet to its neighbors. The neighbors would assign an initial trust value of 0.5 to the node. The trustworthiness of the node can be increased if the node shows benevolent behavior. Similarly, when a node leaves the network, it would no longer respond to the messages. The neighbor may conclude that the network has lost its connectivity or the node has exited the network. In this scenario, the network would delete the node from its network's table and would broadcast this information to other nodes in the network. These nodes would then delete this table from their route cache.

4 Proposed Method

We propose a novel method to enhance security in both phases. We design a secure routing protocol based on trust, which ensures secure and undisrupted delivery of transmitted data. The misbehavior mentioned above can be nullified by securing the data transmitted. An end to end encryption technique is used to self encrypt the data without the necessity of a cryptographic keys. The message is divided into multiple parts, which are self encrypted and forwarded through multiple trust worthy paths between source and destination. In our method, even if an attacker succeeds in receiving one or more transmitted parts, the probability of the original message getting reconstructed is low.

4.1 Multiple Secure Route Discovery:

Multipath routing consists of finding multiple routes between a source and destination node. These multiple paths can be used to compensate for the dynamic and unpredictable nature of ad hoc networks. But such routing protocols are not able to detect and isolate misbehaving nodes and are vulnerable to attack launched by them. So we have designed a multipath secure routing protocol based on the trust information of the node involved. We have modified the traditional route discovery process by embedding the trust information in the RREQ and RREP packets. We discover multiple paths which are node-disjointed. In the node-disjointed paths, nodes on the paths should not be common. Hence, the route discovery mechanism of the existing routing protocol is modified to discover a maximum number of node-disjointed and secure paths. Each node uses less memory, but packet header size is larger because we embed trust information in it.

We introduce the concept of path trust which is derived from the mutual trust value of nodes involved in the path and the total number of nodes. Furthermore, malicious nodes can be avoided from the path as the most trustworthy path is selected.

4.1.1 Path Trust

Path trust is the *trust value* associated with the path. This value is defined as the weighted average of the *trust values* of the nodes in the path. Trust is considered to be asymmetric, so mutual trust between the nodes is used. Hop count also plays a prominent role in the selection of the path since the larger the number of nodes, more is the delay in the network and chances of information modification also increases.

To calculate *path trust*, the RREQ and RREP packets are modified so that they contain the *trust value* of the node from which the packet is received. Both packets are changed because during route discovery a node transmits the RREQ packet by broadcasting. A node knows only the node from which the packet is received, not the node to which it is to be transmitted. Therefore, the RREQ packet is modified to incorporate the previous node's *trust value* and the RREP packet is modified to incorporate the next node's *trust value*.

4.1.2 Route Discovery at Source Node

The source node initiates a route discovery process by broadcasting a RREQ packet. The RREQ packet header is modified by adding a *p trustfield*. *p trust* denotes the trust value of the path up to that node and is initialized as 0 at source node.

$$RREQ : \{IPd;\ IPs;\ Seqnum\}||p\_trust:$$

After broadcasting the RREQ packet, the source node sets a timer whose time period *T* is equal to the 1-way propagation delay and is calculated using formula given below:

$$T = 2 * TR=S + c:$$

4.1.3 RREQ Processing at Intermediate Nodes

An intermediate node is not allowed to reply from its route cache. In our method, an intermediate node forwards the RREQ packet if it received from a different node and itself is node included in the source route of RREQ to avoid route loop. Each RREQ packet is modified to include the trust value of the node from which the packet is received. For example, if there are two nodes A and B in the network, when B broadcasts a RREQ packet and node A receives it, it updates the *p trust* field as:

p trust = *p trust* + *TAB;*

where *TAB* is the trust value that is assigned by node *A* to *B* and signifies how much node *A* trusts *B*. An intermediate node delays the forwarding of RREQ by a time equal to the 1-way propagation delay after receiving the RREQ packet. If the intermediate node overhears a RREP packet with hop count equal to 1 before the timer expires, it and the node that forwarded the RREQ packet are both one hop neighbors of the destination. So the neighbor table is updated.

4.1.4 RREP at Destination Node

The RREP packet header is modified such that it contains two fields' *p trust* and *n trust* in addition to other fields. The updated RREP is:

RREP : *{IPs; IPd; Seqnum}*||*p trust*||*n trust;*

Where *p trust* is assigned from the RREQ packet received at the destination and *n trust* is initialized to 0. It has the same significance as *p trust* in the RREQ packet and denotes the trust value of the path up to that node from the destination.

4.1.5 RREP Processing at Intermediate Nodes

When an intermediate node receives a RREP, it checks if it is the intended next recipient. If yes, it modifies *n trust* in the same manner as *p trust*. For example, when node *X* receives RREP from node *Y* , it updates *n trust* as:

n trust = *n trust* + *TXY :*

The intermediate node forwards the RREP along the route in the source route of RREP. If an intermediate node overhears a RREP and it is not the intended next recipient, then it adds the first node in source route of RREP to its neighbor table. The first node in source route is the one hop neighbor of the destination.

4.1.6 Path Decision at Source Node

When the RREP packet reaches the source node, it calculates *path trust* which is the trust value associated with the path. *path trust* is a weighted average based on the trust values *p trust* and *n trust* received in the RREP packet and the number of nodes in the path as shown in the following equations:

path trusti = ((*path trust* + *n trust*)=2) * *wi;*

where

wi = *n*1=*ni*=X*n*

i=1

1=*nj ;*

path trusts¡d = *max*(*pathtrusti*)*:*

Here *ni* is the number of nodes in *ith* path, *n* is the total number of paths from *s* to *d*, *wi* is the weight assigned to the *ith* path, *path trusti* is the trust value of the *ith* path and *path trusts¡d* is the trust value of the path selected as the most trust-worthy path. Afterwards source node selects *k* node disjoint having *path trust* greater than threshold. Through exhaustive simulation we have set the threshold at 0.6.

To illustrate how to calculate Path trust in DSR Route Discovery, consider the network shown in Figure 1 below. Consider that source node *S* has to send data to destination node *D*. *S* does not have a path to *D*, so it initiates route discovery by sending RREQ to its neighbors. Let the RREQ packet reach node *D* from the path *S ¡ A ¡ E ¡ H ¡ D*. Each intermediate node modifies *p trust* by including the trust value of the node from which it received the packet.

When the RREQ packet reaches node *D*, the value of *p trust* is given by:

p trust = *TAS* + *TEA* + *THE* + *TDH:*

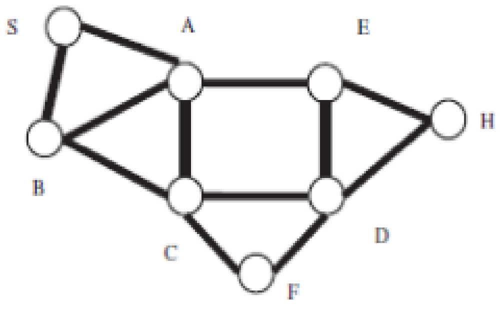

Figure 2: An ad hoc network

Now RREP is sent from node *D* to *S* from the path *D¡H¡E¡A¡S* with *p trust* as in RREQ packet received at *D* and *n trust* initialized to 0. Each intermediate node will update *n trust*. So at *S*, *n trust* will be:

n trust = *THD* + *TEH* + *TAE* + *TSA:*

Therefore *path trusts¡d* = (*p trust* + *n trust*)=2 * *wn* or *path trusts¡d* = ((*TAS*+ *TSA* + *TDH* + *THD* + *THE* + *THE* + *TAE* + *TEA*)=2) * *wn*. Hence *path trusts¡d* contains mutual trust information of all the nodes involved in the path from *S* to *D*. These node disjoint paths are able to detect and isolate misbehaving nodes so they are able to withstand against routing attacks launched by them. Next, we have incorporated end to end data security to withstand against data transmission attacks. These attacks attempt to learn or make use of the information within the network but do not change the data or resources within the system. The release of message contents and traffic analysis are the two primary types of these attacks. They are very difficult to detect because they leave no visible traces. Although the results or the need for securing against these attacks may not be monitored or visibly present, it is still a priority to protect against these seemingly harmless attacks, and more so in the military context.

4.2 End to End Data Security

We have divided the original message into smaller parts and each part is given a unique identifier. Pairs of parts are XOR-ed together, and each pair is sent along a different path. The technique of XOR-ing for encryption is much less compute intensive as compared to public key cryptographic systems [1]. It is therefore very attractive for mobile ad-hoc networks or any application where power consumption and area are important considerations. Information regarding the pair combinations is sent on the most trustworthy path to allow message reconstruction at the destination.

4.2.1 Encryption at Source Node

The message is divided into *k*-1 parts, where *k* is the number of secure and node disjoint routes from source to destination. The most trustworthy path or the path having maximum *path trust* is selected as the indicator path. We generate a random number *x* between 1 and *k*-1, which is used as increment factor for self encrypting message parts.

x = *rand*()%*k:*

The source node informs the destination about the increment factor through the indicator path. Each message part is assigned a unique identifier and is self encrypted using the XOR operation.

The *ith* part of message M is encrypted using an XOR operation as shown below:

p0 i = *pi* © *p*(*x*+*i*¡1)%*k;*

where *pi* is the *ith* part of the message. And *p0i* is the modified *ith* part of the message. The *ith* part of the message is self encrypted by XOR-ing it with *p*(*x* + *i* ¡ 1%*k*) part of the message. The *ith* message is transmitted on the *ith* trustworthy path along the message identifier. The *xth* part of the message is self encrypted by XOR-ing with a random number as shown below:

p0 i = *pi* © *x:*

Since all the parts of the message are self encrypted, an attacker node is not able to get any part of the message. 4.2.2 Decryption at Destination Node When the destination receives the message part along with the identifier, it constructs the message M by decrypting the message part. First of all the *xth* part of the message is discovered as:

pi = *p0*
i © *x;*

where *p0x* is the *xth* encrypted part of the message received. *px* is the original *xth* part of the message. *x* is the increment factor received from the indicator path. Further *ith* part of the message is decrypted as

pi = *p0*
i © *p*(*x*=*i*¡1)%*k:*

4.2.3 Illustration

Suppose in the route discovery phase, five secure node disjoint routes from source to destination are discovered. So the message is divided into four parts *p*1, *p*2, *p*3 and *p*4. The increment *x* has a random value between 1 and 4.

x = *rand*()%4*;*

let *x* = 2. Now the parts of message will be sent as

*p*01 = *p*1*p*((1 + 2 ¡ 1)%4) or *p*01 = *p*1 © *p*2;
*p*02 = *p*2 © *x*;
*p*03 = *p*3*p*((3 + 2 ¡ 1)%4) or *p*03 = *p*3 © *p*4;
*p*04 = *p*4*p*((4 + 2 ¡ 1)%4) or *p*04 = *p*4 © *p*1*:*

5 Security Analyses

We analyze the security of our method in both route discovery and data transmission phases by evaluating its robustness in the presence of some attacks as described below.

5.1 Packet Dropping and Modification

This type of attack involves forging routing packets to cause all routes to go through a misbehaving node. The malicious node then drops or modifies all or some packets for the destination, thus carrying out a black-hole or gray-hole attack, respectively. In our method, such nodes are detected and excluded as the path selected is based on the mutual trust information of the nodes. The trust is quantized and updated based on node behavior. This feature allows the routing algorithm to avoid nodes that are more likely to attempt `breaking-in' the encryption. In addition, suspected nodes which have high computation power and are hence likely to be more successful in crypt-analysis, can be given less parts to stymie their plans.

5.2 Protection Against Malicious Collaborating Nodes

We have used multipath routing for such protection. By using *k* node-disjoint paths of communication, a malicious node should compromise at least *k* nodes and more particularly at least one node in each route, in order to control the communication. According to the operation mode of malicious node, our method offers different levels of protection. In parallel mode, the protocol is resilient against *k*-1 collaborating malicious nodes. In single operation mode, the misbehaving node can disrupt communication by compromising only the active path. The protection of the mixed operation mode lies between the single and parallel mode and may be more efficient for practical applications.

5.3 Traffic Analysis

In existing protocols, a route which is discovered between the source and destination nodes may include a malicious node, which gets to see every packet destined to other nodes. Hence it can analyze the traffic. However, in our protocol, due to selection of trust worthy paths, we are able to detect and exclude misbehaving nodes from the path, thus avoiding the chances of traffic analysis.

5.4 Cache Poisoning

When RREP from the destination is tunneled back to the source node through misbehaving nodes or malicious nodes, then a shorter path is recorded at intermediate nodes and source node, resulting in an attack called cache poisoning. But in our approach we prohibit the RREP from the intermediate node so these nodes do not maintain a route cache. Ultimately space is saved at intermediate nodes and misbehaving nodes are not able to launch this attack.

5.5 Data Security

The requirements of data security in MANETs are basically the same as those defined in traditional networks, that is, data confidentiality, integrity and availability. Data should be accessible only to authorized entities (usually the destination node), should not be corrupted and should be always available upon request to the authorized entities. More specifically, the above three basic requirements can be further elaborated in MANETs as follows.

5.5.1 Data Confidentiality

Unless the attacker can gain access to all the transmitted parts, the probability of message reconstruction is low. This is because, to compromise the confidentiality of the original message, the attacker must listen on all the paths used and decrypt each transmitted part. This is not possible as the message parts are sent through node disjoint paths. So an attacker node will be part of one of the paths and will be able to get only that part of message. Further, since the message parts are self encrypted, it is not possible to extract the original part from the transmitted part of the message.

5.5.2 Data Integrity and Availability

Integrity protects transmitted data from modification, such that only the original source is allowed to write the data. Availability ensures that the data can be successfully transmitted from the source to the destination in a timely manner. In our method data availability and integrity of the message transmitted is enhanced as the path chosen to send the data packets are trust worthy and free from misbehaving nodes. So data loss due to packet drop and packet modification is masked due to removal of such nodes based on the trust values.

6 Results And Discussions

6.1 Simulation

We have used the QUALNET network simulator (Version 4.5) developed by Scalable Network Technologies Inc. [13] to evaluate the effectiveness of the proposed method. Different scenarios are defined in a 1000 * 1000*m* square area with 50 nodes. The source and destination nodes are randomly selected. We used the IEEE 802.11 Distributed Coordination Function (DCF) [3] as the medium access control protocol. A traffic generator was developed to simulate constant bit rate sources. In each scenario, nodes move in a random direction using the random way point model [2] with a speed randomly chosen within the range of 0-20 m/s. The transmission range of each node is 100 m. We assume that there are 0-40% malicious nodes in the network.

6.2 Metrics

To evaluate the performance of the proposed method, under routing phase attacks, we use the following metrics: Route Selection Time. It is defined as the total time required for selecting a path set for routing. Since DSR uses the first path it receives, its path selection time is the time taken in getting the first route reply. Average Latency. It is defined as the mean time in seconds taken by the data packets to reach their respective destinations.

Throughput. It is the ratio of the number of data packets received by the destination node to the number of packets sent by the source node.

6.3 Performance Analysis

Since there is no existing method to enhance security in both the phases of MANETs, we will compare the performance of standard DSR protocol [5], split multipath routing protocol [6], i.e. SMR, trust based multi path DSR protocol [12], i.e. TDSR and our proposed method in the presence of misbehaving nodes. The network is vulnerable to packet drop and modification attack launched by these nodes. The route selection time for all algorithms is presented in Figure 3. Since DSR selects the first path it receives as the path set, the route selection time of DSR is minimum. The disjoint multi-path routing algorithm SMR has to wait for at least two disjoint paths till it can select a path set. TDSR takes the longest time in route selection as it selects the path from all the available paths. Our proposed method takes more time than DSR, since it requires a trusted path to be found. But in cases where all the nodes of the path received first are trusted, the route selection time of our proposed method can equal that of DSR. Hence, we observe that there is a compromise between security and route selection time, which is generally the case with most security algorithms. We have achieved a balance between these two concepts in order to provide maximum security level without causing substantial delay, by choosing the first trust worthy path. However, as trusted protocols endeavor to find the most trusted paths in the network, the selected paths may sometimes deviate considerably from the optimal paths.

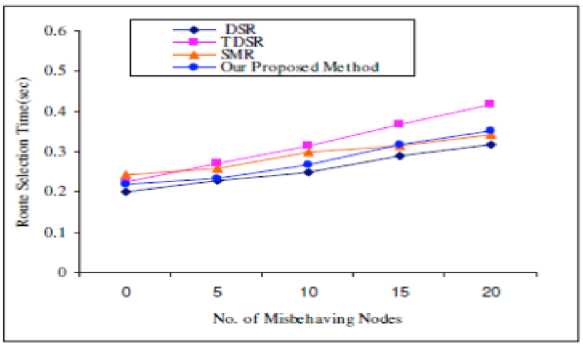

Figure 3: Route selection time

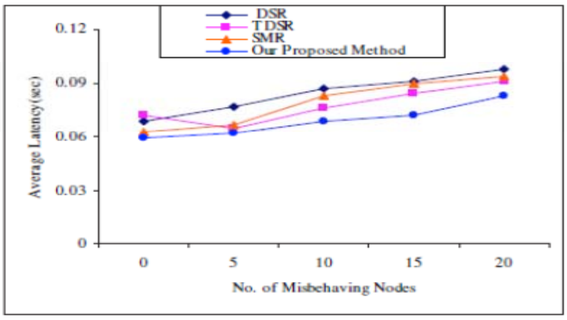

Figure 4: Average latency

This increases the length of the paths, thereby increasing the latency of the network. But the average latency of the network is lower for the multipath protocols compared to DSR, where routing decisions are only made once. Our method has the lowest average latency as shown in Figure 4 because it uses multi path simultaneously and if one of route is disconnected the data is transmitted to next available route.

There is no route acquisition latency. As the number of misbehaving nodes increases, the rate of route recovery also increases, due to the attack launched by misbehaving nodes. So, average latency in DSR and SMR increases significantly. This route recovery is delayed in TDSR and our method as path discovered are trust-worthy.

In our method all the paths are node disjoint, so the impact of misbehavior or link failure is limited only to the specified path. Since data transmission takes place through all the trustworthy paths, data buffering is decreased, which ultimately decreases the average latency.

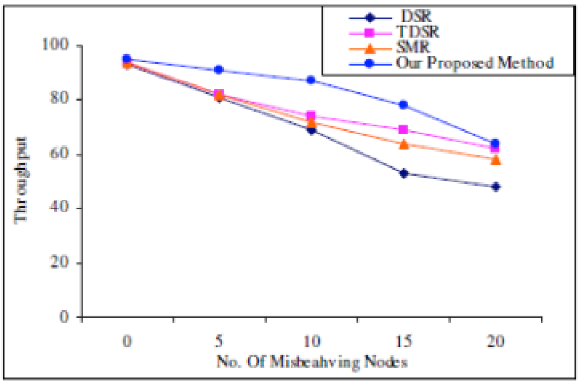

Figure 5: Throughput

When the network is free from malicious nodes, the throughput of TDSR and DSR is the same, but SMR and our method have high throughput due to the multipath feature. Throughput for DSR and SMR degrades steeply with increase in the number of misbehaving nodes in the network, as shown in Figure 5. Throughput of TDSR also decreases with increase of malicious nodes, but is much less compared to DSR and SMR. We have made effective use of the inherent multipath feature selecting path that excludes misbehaving nodes; hence it is able to forward a large number of packets at all traffic loads with minimal loss as seen in Figure 5.

7 Conclusions

In this paper we have proposed a novel method through which we are able to provide end-to-end security in MANETs. Security in the routing phase is enhanced by discovering a secure and trustworthy route through a trust based multipath routing protocol. Multiple node disjoint paths are utilized to enhance the security of the data delivery phase. Misbehaving nodes which induce packet drop or modification attack are detected and exempted from such paths using the trust value of nodes. But sending complete confidential data on the path helps attackers to eavesdrop, whereas sending it in parts on different disjointed paths increases the confidentiality and robustness. This is because it is almost impossible to obtain all the parts of a fragmented message sent on multiple paths existing between the source and the destination. In the worst scenario, even if the attacker node is able to get some parts of the message, it is not possible to deduce any valuable information as these parts are self encrypted. In our proposed method we have secured the transmission between source and destination without the use of cryptographic keys. In addition, the presented encryption and decryption methods are not compute intensive. A limited number of XOR operations are the only requirement. Another advantage is that security is not associated with computational resource requirements as in key encryption systems. We simulated our method and compared its performance with DSR, multipath node-disjoint routing and trust based multipath routing. We have shown that our proposed method is much more secure. It is the only method which is able to withstand against attacks in both phases.

References

[1] L. Batina, S. B. Ors, B. Preneel, and J. Vandewalle, \Hardware architectures for public key cryptography integration," *The VLSI journal*, vol. 34, pp. 1-64, 2003.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. G. Jetcheva, \A performance comparison of multihop wireless ad hoc network routing protocols," *Proceeding of International Conference Mobile Computing and Networking (MobiCom)*, pp. 85-97, 1998.

[3] IEEE Computer Society LAN MAN Standards Committee, *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification*, IEEE Std 802.11-1997, the Institute of Electrical and Electronics Engineers, New York, NY, 1997.

[4] T. Haniotakis, S. Tragoudas, and C. Kalapodas, \Security enhancement through multiple path transmission in ad hoc networks," *IEEE Communications Society*, pp. 4187-4191, 2004.

[5] D. B. Johnson, D. A. Maltz, Y. C. Hu, and J. G. Jetcheva, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)*, Internet draft IETF RFC 3561, 2003.

(http://www.ietf.org/rfc/rfc3561.txt)

[6] S. Lee and M. Gerla, \Split multipath routing with maximally disjoint paths in ad hoc networks," *Proceedings of the IEEE ICC*, pp. 3201-32055, 2001.

[7] W. Lou, W. Liu, and Y. Fang, \SPREAD: Enhancing data confidentiality in mobile ad hoc networks," *IEEE Conference on Computer Communications*, pp. 2404-2413, Hong Kong, China, 2004.

[8] P. Narula, S. K. Dhurandher, S. Misra, and I. Woun- gang, \Security in mobile ad-hoc networks using soft encryption and trust-based multi-path routing," *Computer Communications*, vol. 31, no. 4, pp. 760-769, Mar. 2008.

[9] P. Papadimitratos and Z. J. Haas, \Secure data transmission in mobile ad hoc networks," *International Conference on Web Information Systems Engineering (WISE' 03)*, pp. 41-50, Atlanta, Georgia, USA, 2003.

[10] A. A. Pirzada, A. Datta, and C. McDonald, \Propagating trust in ad-hoc networks for reliable routing," *Proceedings of IEEE International Workshop Wireless Ad Hoc Networks*, pp. 58-62, Finland, 2004.

[11] A. A. Pirzada, A. Datta, and C. McDonald, \Trust based routing for ad-hoc wireless networks," *Proceeding of IEEE International Conference Networks*, pp. 326-330, Singapore, 2004.

[12] N. Pissinou, T. Ghosh, and K. Makki, \Collaborative trust-based secure routing in multihop ad hoc networks," *Networking*, LNCS 3042, pp. 1446-1451, Springer-Verlag, Athens, Greece, 2004.

[13] Poonam, K. Garg, and M. Misra, \Trust based multipath DSR protocol," *Proceedings of Fifth International Conference on Availability, Reliability and Security*, pp. 204-209, Poland, Feb. 2010.

[14] QUALNET simulator.

[15] A. Shamir, \How to Share a Secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[16] C. Wang, X. Yang, and Y. Gao, \A Routing protocol based on trust for MANETs," *Proceeding of Sixth Annual International Conference on Grid and Cooperative Computing*, LNCS 3795, pp. 959-964, Springer-Verlag, Beijing, China, 2005.

[17] C. Yong, H. Chuanhe, and S.Wenming, \Trusted dynamic source routing protocol," *IEEE International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1632-1636, Athens, Greece, 2007.

[18] L. Zhou and Z. J. Haas, \Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. 1-12,1999.