International Journal of Science and Engineering Applications (IJSEA)
Volume 2 Issue 4, 2013, ISSN - 2319-7560 (online)

Security Establishment in MANET Systems using FACES

P.Manivannan
Department of Information Technology
V.R.S. College of Engineering and Technology
Arasur, Villupuram Dt – 607107,
TN, India.

J.Shakila
Department of Information Technology
V.R.S. College of Engineering and Technology
Arasur, Villupuram Dt – 607107,
TN, India.

Abstract: Friend based Ad-hoc routing using Challenges to Establish Security (FACES) is an algorithm to provide secure routing in ad-hoc mobile networks. The scheme proposed has been drawn from a network of friends in real life scenarios. The algorithm is divided into four stages, viz. Challenge your neighbour, Rate Friends, Share Friends and Route through Friends. One of the major advantages of this scheme is that the nodes do not need to promiscuously listen to the traffic passing through their neighbours. The information about the malicious nodes is gathered effectively by using Challenges, which reduces the overhead on networks. As a result of this scheme of operation, the network is able to effectively isolate the malicious nodes which are left with no role to play in the ad-hoc network. One major benefit of this scheme is that the nodes do not need to promiscuously listen to the traffic passing through their neighbours. The information about the malicious nodes is gathered effectively by using Challenges. This reduces the overhead on the network significantly. Through extensive simulation analysis it was inferred that this scheme provides an efficient approach towards security and easier detection of malicious nodes in the mobile ad-hoc network.

Keywords: FACES, Challenges, Rate friends, Share Friends, Net Friends

1. INTRODUCTION

Wireless technologies have revolutionized the world of communications. It started with the use of radio receivers or transceivers for use in wireless telegraphy early on and now the term *wireless* is used to describe technologies such as the cellular networks and wireless broadband Internet. Although the wireless medium has limited spectrum along with a few other constraints as compared to the guided media, it provides he only means of *mobile communication*. Wireless ad-hoc networking is used for random and rapid deployment of a large number of nodes, which is a technology with a wide range of applications such as tactical communications, disaster relief operations, health care and temporary networking in areas that are not densely populated. A mobile ad-hoc network (MANET) [1] - [3] consists of mobile hosts equipped with wireless communication devices. The transmission of a mobile host is received by all hosts within its transmission range due to the broadcast nature of wireless communication and omni-directional antenna. If two wireless hosts are not within the transmission range in ad-hoc networks, other mobile hosts located between them can forward their messages, which effectively build connected networks among the mobile hosts in the deployed area.

1.1 Objective

In this paper, we present the design and propose an algorithm to establish secure routing in mobile ad-hoc networks. We name the algorithm as FACES which stands for Friend based Ad-hoc routing using Challenges to Establish Security. The name of the algorithm itself explanatory. We use trust establishment through friends and special challenges for authenticating the nodes. This provides a robust mechanism for thwarting attacks by isolating malicious nodes in the network. We also propose friend updating schemes and suggest a novel way to authenticate nodes using challenges, which is the basis of the algorithm. The algorithm tackles all the security challenges in an innovative way and gives a robust self-sustaining security mechanism without data is finally routed through the route with the greatest number of trusted friends. The quality of the route is determined by evaluating each and every node in the route and making a final decision about the quality of the route. To deal with eavesdropping we encrypt the data at the source using *public key cryptography*. A central authority such as a key distribution center can be very difficult to maintain in a mobile ad-hoc network. So, whenever a destination node receives a route request it sends its public key along with the route reply. The source uses that public key, which it receives from the most trusted route to encrypt the data that needs to be sent. In this way the chances of man in the middle attack are greatly reduced and eventually are eliminated as the friend circle becomes much more robust. The use of wireless ad-hoc networks also introduces additional security challenges that have to be dealt with. The weak links that cause these security challenges are as follows.

1.1.1 Easier to Tap

Since the media is nothing but air, it can be tapped easily.

1.1.2 Limited Capacity

The wireless medium has limited capacity and therefore requires more efficient schemes with less overhead.

1.1.3 Dynamic Nature

The self-forming, self-organization and self-healing algorithms required for ad-hoc networking, may be targeted to design sophisticated security attacks.

1.1.4 *Susceptibility to Attacks*

The wireless medium is more susceptible to jamming and other denial-of-service attacks. Attacks in MANETs can be broadly classified as: passive and active attacks. In passive attacks the intruder remains undetected and captures the data while the message is being transmitted over the network. Eavesdropping and traffic analysis mainly fall in this category. Unlike passive attacks, in active attacks the intruder/attacker can affect the communication by modifying the data, misleading the nodes in the network. As a matter of act various scenarios and threats can be developed based on these approaches.

2. RELATED WORK

This section discusses the previous work done in the field of secure routing in ad hoc networks. The goals of any secure routing protocol are to provide some or all of the properties such as Authentication, Access Control, Confidentiality, Privacy, Integrity, Authorization, Anonymity, Non-repudiation, Freshness, Availability, Resilience to attacks. Of these, Availability in particular targets denial of service (DoS) [5] attacks and has the ability to sustain the networking functionalities without any interruption due to security threats. The routing algorithms deal with the dynamic aspects of Mobile Ad-Hoc Networks in their own way depending upon the requirements of the system. Essentially a routing algorithm can behave in a reactive, proactive, or a combination of both, that is, in a hybrid way. Reactive algorithms are those that behave in an on-demand fashion, which means that these algorithms gather routing information in response to some event viz. start of a data session, route request messages, link failure messages etc. Proactive algorithms are those which gather essential information before hand, so that the information is readily available when an event occurs. Hybrid algorithms use both proactive and reactive components in order to try to combine the best of both schemes. The conventional routing protocols for MANETS are DSR [4] and AODV [6]. These conventional routing algorithms do not provide security and are prone to attacks caused by malicious nodes moving in the network. Since security is one of the major concerns of ad-hoc networks there is a need for secure routing schemes in ad-hoc networks. This can be achieved by using either of the following security based routing methods: payment-based systems, reputation-based systems and cryptography-based systems. All these systems have their own features. Of these, the reputation-based systems and the cryptography-based systems are the ones that are most widely used in ad-hoc networks. It has also been observed that most of the secure routing algorithms use cryptography as the central mechanism to implement security. Two of the most widely used algorithms for public key cryptography are RSA and Diffie – Hellman [7], [8]. A number of routing protocols [9] - [16] have been proposed towards providing security in ad-hoc networks. Some of the most widely discussed protocols are Authenticated Routing for Ad Hoc Networking (ARAN) [9], ARIADNE [10] and Watchdog Pathrater [11]. There have also been various secure routing techniques [14] - [16] that use multipath based routing where they break the data into different number of sub packets, encrypt them and then finally route them through different paths. In this work we have looked into the secure routing techniques DMR [14], TMR [15] and MTMR [16], and have designed the proposed FACES protocol to provide better security. These protocols [14] - [16] have been discussed in the following subsections, as these protocols are the ones that have been used for comparison with the proposed technique FACES.

2.1 Security Enhancement through Disjoint Multipath Transmission: DMR

DMR [14] provides a way to further secure the data transmitted along routes of a wireless ad hoc network after a potentially secure connection has been established between two nodes. In this method, the encryption/decryption key used is the message itself. The approach requires that the message is split into parts (sub-messages) and that the encrypted sub-messages be transmitted along different paths (routes) which are reception disjoint. The method partitions a 4n-bit message into two four n-bit parts called. Up to three redundant bits can be added in order to make the number of bits a multiple of four. Four encrypted n-bit parts, labeled are generated using the equations referred in [14]. For details regarding the encryption and decryption of the message, refer the technique discussed in [14]. This protocol takes advantage of the shortest path between the source and the destination. A modification of Dijkstra’s algorithm is applied for this purpose. All nodes have positive weight. Every path that is returned and is a desired path automatically implies that a second path exists with the reception disjoint property. This set of paths will be used as the solution to the routing problem. Another feature used to determine the security of each selected route is the “priority” labeling. The nodes are labeled with a “priority” number according to the number of edges they are linked to. This indicates if a node can be trusted on sending a message with less chance of that message being grabbed by an adjacent node. In this method, the decryption of the original message requires all the encrypted parts. The security of this method lies in the fact that an enemy node or a corrupted node needs to intercept all the parts to be able to decipher the message. Failure to intercept one part gives no information about the original message.

2.2 Message Security using Trust-Based Multipath Routing : TMR

TMR [15] provides a method of message security using trust based multipath routing. In this approach, less trusted nodes are given lesser number of self-encrypted parts of a message, thereby making it difficult for malicious nodes to gain access to the minimum information required to break through the encryption strategy. Using trust levels, it makes multipath routing flexible enough to be usable in networks with “vital” nodes and absence of necessary redundancy. In addition, using trust levels, it avoids the non-trusted nodes in the routes that may use brute force attacks and may decrypt messages if enough parts of the message are available to them. This technique uses a variation of the trust models used in [17] and [18]. A node is assigned a discrete trust level in the range of to 4. A trust level of 4 defines a complete trust and a trust level of defines a complete distrust. These trust levels also define the maximum number of packets which can be routed through those nodes. 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 acknowledgements received. The 4 n-bit message is divided into 4 n-bit parts, and encrypted using the equations referred in [15].The encrypted parts are then routed instead of the original message using multiple paths between the source and the destination nodes. These multiple paths between the source and the destination nodes are found using DSR. In this, the source node waits for a predefined time period in order to have multiple paths to the destination. The routing paths are finally selected from the set of obtained paths using a novel trust defined strategy in which a node with a trust level of is given at most parts of the packet to forward. This limits the possibility of a brute force decryption of the message. The routes are selected using a greedy approach on the basis of path length such that a node with a trust level of does not get more than packets on the route to the destination. At the destination, the message parts are then decrypted using the equations referred in [15].Thus, the TMR approach is found to be more secure than the multipath routing using disjoint paths (DMR), but it generally takes more time in route selection.

2.3 Message and Trust Based Multipath Routing: MTMR

MTMR [16] uses a trust assignment and updating strategy which can be used to identify and isolate malicious nodes without being hard on the resources of the network. It uses a parameter, the trust requirement of the message such that each message has a certain level of importance based on its content and type. This is the trust requirement of a particular message, which decides how the message will be routed. Therefore, only paths with certain trust level can be used for its forwarding. This further enhances the security of the system. Initially, each node is given a trust value of zero which indicates unknown trust level. Later this value may be incremented or decremented based on the behavior of the node. The trust levels have a range of values from for minimum trust and for maximum trust. Equations (1) and (2) are used for decrementing and incrementing the trust of a node. In these equations indicates the allowed number of misbehaviors that a node with a given value of trust can perform and, indicates the number of times a normal behavior was exhibited by a node with trust (1) & (2). If the trust is calculated as 1, then the value of will be equal to 2. Therefore, it will take two misbehaviors to reduce trust value to an immediate lesser trust level of 0. Similarly, for a node with trust level 4, eight (8) misbehaviors are allowed, also if a node with a current trust value as 3 has to rise to a trust level of 4, it will have to perform normally for at least 8 times. Equation (3) below (based on the technique referred in [19]) calculates the trust value of a given node by its neighboring nodes. In this equation, is the trust level that the node has of node is the trust level that node has observed on node , and represents the required trust level for the current message delivery.(3) The MTMR approach uses the message encryption inspired by cipher-block chaining (CBC) mode of block encryption referred in [20], [21]. It defines a trust based path selection strategy where a path with trust is given only parts of the packet to forward. This limits the possibility of brute-force decryption of the message by any node with lower trust value than the message. The multiple paths are calculated by DSR, by waiting for a specified period of time for the multiple *Route_Reply* packets to come from various paths. The paths are then arranged in an ascending order of hop-counts and descending order of trust levels. This step makes sure that the routes selected are of least hop-count besides being most trusted, so as to minimize the overheads and the path with highest trust is selected. Once the paths have been selected, the parts of the data packets are then transmitted through these selected paths based on the routing decisions discussed in [16]. Once the parts of the packets have been sent completely, the source then sends the hash of the complete packet as the final message. The hash message is calculated as a Cyclic Redundancy Check (CRC) variant [20].

3. FACES PROTOCOL

In this section, we discuss our proposed algorithm in detail. We start with the list of terms used in the protocol. This is followed by a detailed discussion of the algorithm and a list of security attacks thwarted by it.

3.1 List of Terms Used

3.1.1 Question Mark List

The list of nodes which are deemed suspicious by a particular node. This list is stored for each and every node in its data structure.

3.1.2 Unauthenticated List

The list of nodes of which no security information is present.

3.1.3 Friend List

This is the list of nodes which convey trust. Like the question mark list, a friend list is also stored for each node in its data structure. Friends are rated on a scale of 0 to 10.

3.1.4 Friend Request (FREQ)

This is a control packet which is used to initiate friend sharing. A node receiving this packet replies with the nodes in its friend list, unauthenticated list and the question mark list.

3.1.5 Data Rating (DR)

This is the rating given to nodes after they transmit some amount of data for the source node.

3.1.6 Friend Rating (FR)

This is the rating computed when nodes share their friend lists.

3.1.7 Net Rating (NR)

This rating is computed as a weighted mean of DR and FR.

3.1.8 Obtained Rating (OR)

The rating received during the friend sharing stage.

3.2 FACES Algorithm Description

Friend based Ad-hoc routing using Challenges to Establish Security (FACES) accomplishes establishment of friend networks in MANETs in the same way as in real life scenarios. We apply the same idea to develop the FACES algorithm. The proposed FACES algorithm is divided into the following four stages as shown in Figure 1(a) & 1(b) - *Challenge your neighbor, Rate Friends, Share Friends and Route through friends*.

The figure 1(b) also depicts the link/flow between the different stages of the algorithm. The routing of data in the protocol is on demand; that is whenever the need arises. But challenges, friend sharing and rating are periodic processes. This makes the FACES protocol a hybrid one. The *Challenge your neighbor* stage is designed to facilitate trust establishment for a new node in relation to the other nodes present in the network. *Rate Friends, Share Friends and Route through friends* gradually make the network robust in terms of the reliability of the nodes, and it is through these stages that the nodes gather data about each other and populate a *friend list* where the information about reliable nodes is kept. A node having its neighbors in its friend list does not need to challenge them before a data session. The idea of the FACES scheme is drawn from real life friend networks. When people meet in a new community or a group they are strangers to each other. Fig. 1(a) depicts a network of friends in a community. Tasks are completed by trusting one another unconditionally initially and with time the trust level increases with the number of successful task completions. Initially breach of trust is possible as no one has any information about the people with malicious intentions. However, with time, trust relationships are formed and we have a community where tasks are completed efficiently. The following sections discuss each of the stages in greater detail.

3.2.1 Challenge your neighbor

Challenge is a mechanism to authenticate nodes initially when no criterion is present. It is a basic test which a node has to complete in order to prove its honesty and integrity. Let us assume that the node challenges its neighbor node.

Step 1) When the network is newly initialized, each node is a stranger to another. Thus each node incorporates its neighbors in the *unauthenticated list*.

Step 2) The node picks one of the neighbors, and performs the usual *Share Friends Stage* (which will be discussed later).

Step 3) As a response the neighbor node either sends its friend list or the nodes from its *unauthenticated list* if the friend list is empty.

Step 4) On receiving the list, the node picks up a node which it can reach on its own and in the most efficient way. Let us say that this node is.

Step 5) Now the node has two ways to reach the node one through and another through a route already known to it.

Step 6) The node initiates a challenge and encrypts it with the public key of. It then sends it through both routes also includes its own public key with the challenge.

Step 7) The node sees the challenge as a normal data packet and routes it.

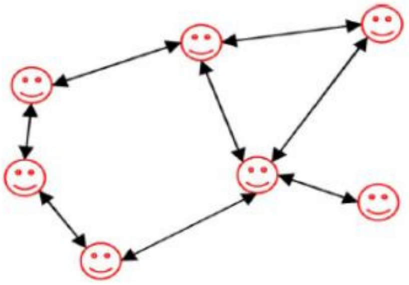


Figure.1(a) - Network of friends in a community

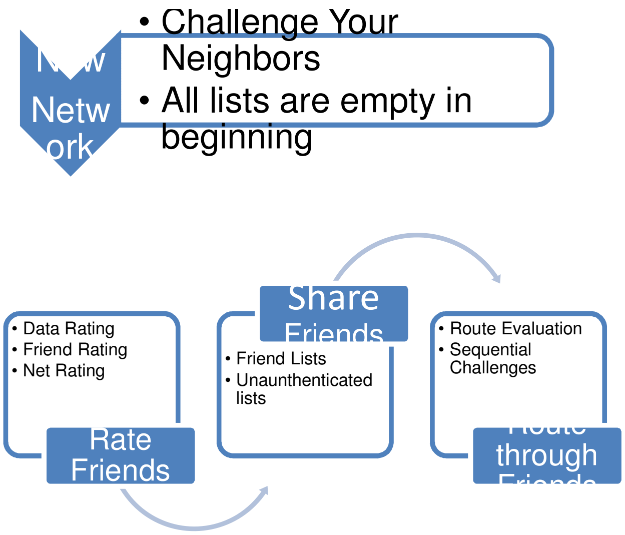


Figure.1(b) - FACES : Link / Flow between different stages

As decrypts the data packet and finds that it is a challenge it responds to the challenge. It then encrypts the response with public key that it obtained in STEP 6.

Step 8) Receives the result of the challenge from both routes and after decrypting, it compares them. If they are same then node adds node at the bottom of its friend list. In this way, node can authenticate node as a node which is behaving genuinely at least initially.

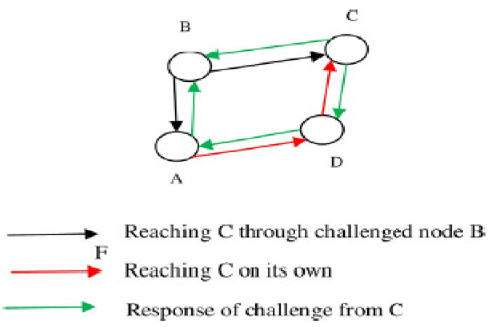

B
C
A
D
F
Reaching C through challenged node B
Reaching C on its own
Response of challenge from C

Fig. 2. Illustration of the Challenge.

Figure.2 gives an illustration of how the challenge is initiated by on disguised as a data packet for. The challenge is also routed through to Figure 2. Illustration of the challenge and the results obtained are compared to arrive at a decision about the node .However, there are some cases which might bring ambiguity in the mind of the reader. We discuss these cases though the method of questions. Each question is formulated to depict the working of this stage of the algorithm. Below, we provide the questions and the answers for the issues which test the suitability of the *Challenge Your Neighbor* Stage.

3.2.2 Description of the challenge

Each node is initialized with a pair of large prime integers which is secret to that node. When a node wants to send a challenge to a particular node it sends one of his random prime numbers to it and expects a response in return. The challenge process takes the following four steps for node challenging node.

Step 1) First it is initialized with

Step 2) When challenges, as described above. It sends a random prime number “n” as the challenge.

Step 3) computes *mod n* and sends the result to the two paths.

Step 4) compares the result from the two paths to arrive at a decision on as described above. Since and are all very large prime numbers it is impossible to determine and from the result of the *mod* function as that is known to be a hard problem. In this way, the nodes can authenticate each other through the challenge process. As the newly initialized nodes authenticate each other and a robust network of friends is formed, it becomes very difficult for a new malicious node to authenticate itself.

3.3 Rate Friends

Friends are rated on a scale of zero to ten. Initially each node has only those nodes in their friend list that completed the challenge successfully. Sharing of friend nodes is done in the *Share Friends* stage as the friend relation is transitive in nature that is a friend of friend includes in his friend list too. Each friend in the list has the following three classes of ratings: *Data Rating (DR)*, *Friend Rating (FR)* and *Net Rating (NR)*.

3.3.1 Data Rating

The data rating is updated by a node for its friend on the basis of amount of data it transfers for it. This is a significant metric for judging the quality of the node, as it portrays its battery power and general capacity to forward data packets. The DR of a friend node varies according to the number of data packets transferred through it. The net DR is calculated as a moving average of the last five data ratings. Equation (4) describes the moving average relation between a data rating and the previous five data ratings:(4) The DR for a particular session is calculated as (5) where is the number of data packets transmitted and is the factor by which we want the number of data packets to be related to the rating. The moving average is a significant tool to estimate the recent quality of node in terms of data forwarding. As and when a node drops data packets, we compute the negative value for one session of DR using as the number of data packets dropped. The exponential scaling on the number of data transferred is an effective tool to scale according the requirements of the network. We can change the value of according to the volume of data that is transferred trough the network. Keeping a value (of 1/100) ensures a smooth scaling from 1 to 10 for data packets up to 200 with a data rating of around 6 for 100 packets. As we increase the value of, the curve increases DR quickly towards the maximum value 10. As is decreased it smoothens the DR along the range of the data packets. Fig. 3 below shows the graph of DR versus the number of data packets transmitted up to 200 with.

3.3.2 Friend Rating

During the Friend Sharing stage a node asks for the friend list of node and incorporates the rating of friends in the following way.

1) If a node have common friend, then the node obtains the rating of the node from node as (6), shown at the bottom of the page. The idea behind this (6) is to incorporate the trust that node has on node while obtaining the rating of node from it. We further explain this through the use of two scenarios.

3.3.3 Net Rating

The idea behind calculating DR and FR is to have two opinions in front of each node. This is done because malicious nodes can identify some nodes for which they would work properly while for some they would drop packets. The DR acts as the soul opinion of the host node and FR acts as the opinion of its friend nodes. The Net Rating (NR) would be a weighted mean of the two ratings as given in equation (1):

$$NR = \frac{W_1 * DR + W_2 * FR}{W_1 \ + W_2} \qquad \text{------------ } \mathbf{(1)}$$

Where *W1* and *W2* would be the weights assigned to DR and FR respectively. The values of *W1*and *W2* are network dependent and can be learnt with experience.

4. SIMULATION RESULTS AND DISCUSSIONS

Figure.3 - Number of Hops

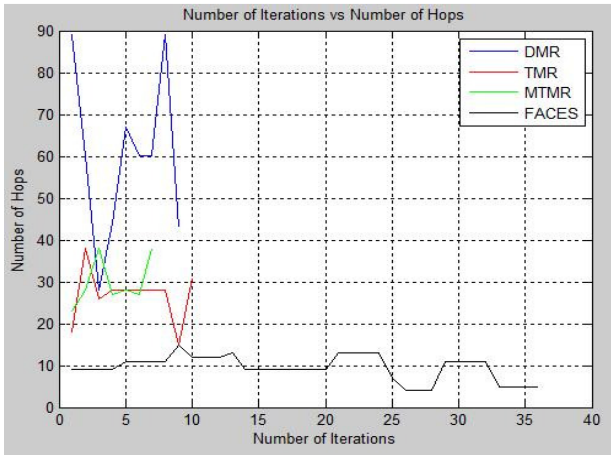


Figure.4 - Route Discovery

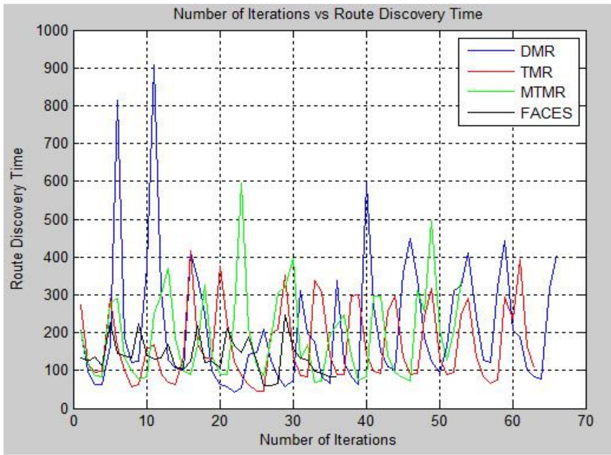
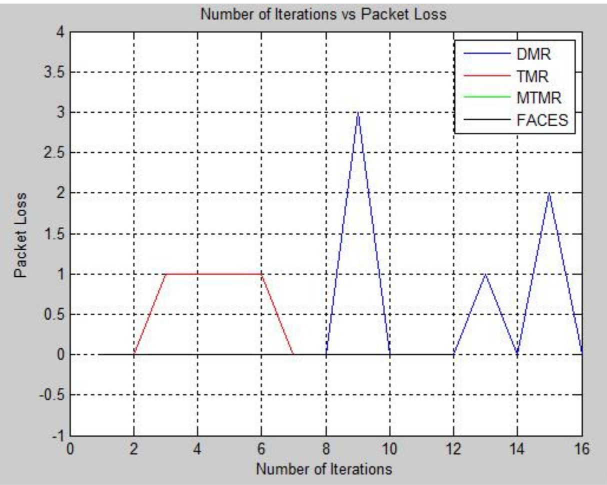


Figure.5 - Packet Loss



5. CONCLUSION AND FUTURE WORK

Mobile Ad-hoc network (MANETs) due to its dynamic nature has many challenges. Some of the major challenges are number of malicious nodes detected, number of hops, route discovery time, packet loss, energy and power consumption.

Many Routing algorithms namely DMR, TMR, MTMR and FACES have their own way in order to establish the trust and transmit packet securely. But Friend based protocol proved to be best in terms of number of malicious nodes detected, number of hops, route discovery time, packet loss, power consumption and energy.

After a logical analysis and extensive simulation of the FACES algorithm under different scenarios, we come to the conclusion that it offers robust scheme to afford security for mobile ad-hoc networks and performs better than the trust based protocols from which it was compared. Due to the absence of the need of promiscuous mode in the mobile nodes, the network has to bear a lot less overhead as compared to other secure routing schemes. The friends sharing scheme turns out to be an efficient mechanism to spread information about trusted nodes effectively in the system. In our protocol, we use challenges to authenticate any node compared to the other security protocols that use multipath routing and overhear the neighbor activities. To make a decision that a node is malicious, the multipath routing algorithms take much more time than FACES scheme which detects the malicious activity by checking the challenge reply. This on the other hand reduces overheads and hence reduces the chances of unsecured routing through faulty nodes. Due to these challenges, the FACES protocol works much better and provides more security than the other multipath routing protocols. In the future, we plan to implement existing secure routing protocols such as the ARIADNE and ARAN and compare them with the proposed FACES protocol. This would give a better picture about the standing of the FACES algorithm as compared to these long established secure routing protocols for MANETs.

6. REFERENCES

[1] D. P. Agrawal and Q.-A. Zeng*, Introduction to Wireless and Mobile Systems*. Pacific Grove, CA: Brooks/Cole, Thomson, 2002.

[2] I. Chlamtac, M. Conti, and J.-N. Liu, “Mobile ad-hoc networking: Imperatives and challenges,” in *Ad-Hoc Networks*. New York: Elsevier, 2003, vol. 1, pp. 13–64, No. 1.

[3] L.Wang and N.-T. Zhang, “Locally forwarding management in ad-hoc networks,” in *Proc. IEEE Int. Conf. Communications, Circuits and Systems and West Sino Expositions*, Jun./Jul. 2002, pp. 160–164.

[4] D. Johnson and D. Maltz, “Dynamic source routing in ad-hoc wireless networks,” in *Book Chapter in Mobile Computing*, T. Imielinskiand H.Korth, Eds. Dordrecht, The Netherlands: Kluwer, 1996, pp. 131–181.

[5] A. Wood and J. A. Stankovic, "A taxonomy for denial-of-service attacks in wireless sensor networks," in *Handbook of Sensor Networks:*ll

[6] C. Perkins, E. Royer, and S. Das, Ad-Hoc on Demand Distance Vector (AODV) Routing Jul. 2003, Internet experimental RFC 3561.

[7] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[8] M. S. Obaidat and N. Boudriga*, Security of e-Systems and Computer Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[9] K. Sanzgiri, B. N. Levine, C. Shields, B. Dahill, and E. M. Belding-Royer, "A secure routing protocol for ad-hoc networks," in *Proc. 10thIEEE Int. Conf. Network Protocols (ICNP)*, Paris, France, Nov. 12–15,2002, pp. 78–89.

[10] Y. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad-hoc networks, "*WirelessNetw.*, vol. 11, no. 1–2, pp. 21–38, Jan. 2005.

[11] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. MobiCom 2000*, Boston, MA, Aug. 2000, pp. 255–265.

[12] M. G. Zapata and N. Asokan, "Securing ad-hoc routing protocols," in *WiSe'02: Proc. of 1st ACM Workshop on Wireless Security*, Atlanta, GA, Sep. 28, 2002, pp. 1–10.

[13] P. Papadimitratos and Z. J. Haas, "Secure link state routing for mobile ad-hoc networks," in *IEEE International Symposium on Applications and the Internet-Workshop on Security and Assurance in Ad-Hoc Networks*, Orlando, FL, Jan. 2003, p. 379.

[14] T.Haniotakis, S. Tragoudas, and C. Kalapodas, "Security enhancement through multiple path transmission in ad-hoc networks," in *2004 IEEE Int. Conf. Communications*, Jun. 2004, vol. 7, pp. 4187–4191.

[15] P. Narula, S. K. Dhurandher, S. Misra, and I. Woungang, "Security in mobile ad-hoc networks using soft encryption and trust based multipath routing," *Sci. Direct Comput. Commun.*, vol. 31, pp. 760–769, 2008.

[16] S. K. Dhurandher and V. Mehra, "Multi-path and message trust-based secure routing in ad-hoc networks," in *Proc. Int. Conf. Advances in Computing, Control and Telecommunication Technologies (ACT 2009)*, Trivandrum, India, Dec. 28–29, 2009, pp. 189–194.

[17] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proc. 1997Workshop on New Security Paradigms*, Langdale, Cumbria, U.K., Sept. 23–26, 1997, pp. 48–60, NSPW'97. ACM Press, New York.

[18] A. A. Pirzada, A. Datta, and C. McDonald, "Propagating trust in ad-hoc networks for reliable routing," in *Proc. 2004 Int. Workshop on Wireless Ad-Hoc Networks*, May–Jun. 2004, pp. 58–62.

[19] Z. Liu, A. W. Joy, and R. A. Thompson, "A dynamic trust model for mobile ad-hoc networks," in *Proc. 10th IEEE Int. Workshop on Future Trends of Distributed Computing Systems*, May 2004, pp. 80–85.

[20] W. Stalling, "Cryptography and network security, principles and practice," in *Book Chapter in Block Ciphers and Data Encryption Standard*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 2006, pp. 62–94.

[21] M. S. Obaidat and N. Boudriga*, Security of e-Systems and Computer Networks*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[22] P. J. Rousseeuw and C. Croux, "Alternatives to the median absolute deviation," *J. Amer. Statist. Assoc.*, vol. 88, no. 424, pp. 1273–1283, Dec. 1993.