International Journal of Advance Research in Science and Engineering
Vol. No.4, Issue 08, August 2015
www.ijarse.com
IJARSE
ISSN 2319 - 8354

SHORTEST SECURE ROUTE OVER MANET

Sakshi Wadhwa[1], Anu[2], Ankush Chhabra[3]

[1]M.Tech, [2]Asst. Professor, CSE, S.K.I.E.T, K.U.K, (India)

[3]Asst. Professor C.S.E), M.M.U,Ambala, (India)

ABSTRACT

A Mobile Ad hoc network (MANET) is a system of wireless mobile nodes that dynamically self-organize in arbitrary and temporary network topologies. In MANET, nodes can directly communicate with all the other nodes within their radio ranges; whereas nodes that not in the direct communication range use intermediate node(s) to communicate. All the nodes that have participated in the communication automatically form a wireless network, and can be viewed as mobile ad hoc network. Because of these wireless features the MANET are more prone to suffer from the malicious behaviors than the traditional wired networks. Therefore, we need to pay more attention to the security issues in the mobile ad hoc networks. In this paper, we have proposed GDH (GENERAL DIFFIE-HELLMAN) to find out the shortest path and then imply the security in multicast routing. The routing is done with the help of GDH, Encryption and Decryption of data. It will also improve the performance of the network such as delay and packet delivery.

Keywords: ***MANET, Wireless Networks, Ad hoc Networking, Routing Protocol.***

I. INTRODUCTION

Mobile Ad Hoc Networks (MANETs) has become one of the most prevalent areas of research in the recent years because of the challenges it pose to the related protocols. MANET is the new emerging technology which enables users to communicate without any physical infrastructure regardless of their geographical location, that's why it is sometimes referred to as an ―infrastructure less‖ network. The proliferation of cheaper, small and more powerful devices make MANET a fastest growing network. An ad-hoc network is self-organizing and adaptive. Device in mobile ad hoc network should be able to detect the presence of other devices and perform necessary set up to facilitate communication and sharing of data and service. Ad hoc networking allows the devices to maintain connections to the network as well as easily adding and removing devices to and from the network. Due to nodal mobility, the network topology may change rapidly and unpredictably over time. The network is decentralized, where network organization and message delivery must be executed by the nodes themselves. Message routing is a problem in a decentralize environment where the topology fluctuates. While the shortest path from a source to a destination based on a given cost function in a static network is usually the optimal route, this concept is difficult to extend in MANET. The set of applications for MANETs is diverse, ranging from large-scale, mobile, highly dynamic networks, to small, static networks that are constrained by power sources. Besides the legacy applications that move from traditional infrastructure environment into the ad hoc context, a great deal of new services can and will be generated for the new environment. MANET is more vulnerable than wired network due to mobile nodes, threats from compromised nodes inside the network,

limited physical security, dynamic topology, scalability and lack of centralized management. Because of these vulnerabilities, MANET is more prone to malicious attacks.

II. RELATED WORK

The field of Mobile Ad hoc NET works is one of the most advanced and rapidly evolving fields in computer sciences and is very vast. Lots of researchers are innovating and bringing new ideas for its development. This section covers the research papers, literatures and journals that were surveyed.

Bashir Yahya, Jalel Ben-Othman, ["Achieving host mobility using DNS dynamic updating protocol", Proceedings of LCN 2008, Pages: 634-638, Montreal, Canada], proposed a distributed naming system based on the peer-to-peer Chord protocol which provides naming resolution in the presence of mobility and node failures. The Mobile Ad hoc Networks (MANETs) are formed by collection of, potentially mobile, portable devices without established infrastructure. In such networks, connectivity is established through IP addresses and dynamic routing. MANETs are highly dynamic with frequent nodes joining and departure. Due to this node dynamicity, the node's IP address changes frequently. Therefore, to enable and run common user applications (e.g. web browsing and email) on such networks, the use of well known names that are unique and easy to remember is required. Thus a name resolution mechanism which maps names to their corresponding IP addresses has become necessary which is a difficult task and the use of the traditional DNS system becomes impossible in such networks.

Bongsoo Kim, Younghwan Choi, Kwansoo Jung, Hochoong Cho, Fucai Yu, and Sang-Ha Kim["A Dynamic Single-Hop Clustering Mechanism Adapted to Overlay Multicast in Mobile Ad hoc Networks" Dept. of Computer Engineering, Chungnam National University, 220 Gung-dong, Yuseog-gu, Daejon, Korea, 369-764], discussed the overlay multicast in mobile ad hoc networks (MANETs) with densely placed member nodes can cause inefficient packet transmission. Dynamic Single-Hop Broadcast Cluster (SHBC) is proposed to solve the problem of overlay multicast. SHBC is a clustering mechanism adapted to overlay multicast in MANETs. SHBC is based on LBC (Local Broadcast Cluster) proposed in Efficient End System Multicast for Mobile Ad Hoc Networks but only constructs clusters in the highly dense areas.

R. Lin and M. Gerla, ["Adaptive clustering for mobile wireless networks", IEEE Journal on Selected Areas in Communications, 15(7), pp. 1265-1275, Sepember 1997] presented Adaptive Clustering architecture for multimedia support in a multihop mobile network. The nodes are organized into non overlapping clusters. The clusters are independently controlled and are dynamically reconfigured as nodes move. This network architecture has three main advantages. First, it provides spatial reuse of the bandwidth due to node clustering. Secondly, bandwidth can be shared or reserved in a controlled fashion in each cluster. Finally, the cluster algorithm is robust in the face of topological changes caused by node motion, node failure and node insertion/removal. Simulation shows that this architecture provides an efficient, stable infrastructure for the integration of different types of traffic in a dynamic radio network.

III. PROPOSED WORK

In the normal data transfer if we send data from source to destination there can be many routes. Out of them some paths are small and some are large and they will consume a lot of time and cost of data transfer. So it will

be a bad idea to send the data directly from source to destination. That is why we found the shortest path first and then decides the rout. Distributed BELLMAN-FORD algorithm is applied to find the shortest path. After that for secure data transfer we do the multicast routing and the routing is done with help of GDH algorithm. The main problem is that there can be an intruder (malicious node) in between the path of source and destination node. This intruder can receive the data from the source node and alter the data before pass it to the destination node by intrudes the key shared between source and destination. So the destination node is unable to know that the data coming is secure or altered by some malicious node .So it was the big issue of secure data transfer in the mobile ad hoc networks. To resolve this problem we made a pool of authenticated nodes. By using this pool of node it is easy to find out the malicious node and a secure data can be send.

3.1 Distributed Bellman-Ford

The DBF algorithm was developed originally to support routing in the ARPANET. A version of it is known as Routing Internet Protocol (RIP) [1] and is still being used today to support routing in some Internet domains. It is a table-driven routing protoco1, that is, each router constantly maintains an up-to-date routing table with information on how to reach all possible destinations in the network. For each entry the next router to reach the destination and a metric to the destination are recorded. The metric can be hop distance, total delay, or cost of sending the message. Each node in the network begins by informing its neighbors about its distance to all other nodes.

The receiving nodes extract this information and modify their routing table if any route measure has changed. For instance, a different route may have been chosen as the best route or the metric to the destination may have been altered. The node uses the following formula to calculate the best route:

$$D(i, j) = \min [d(i, k) + D(k, j)]$$

Where $D(i,j)$ is the metric on the "shortest" path from node ***i*** to node j , $d(i, k)$ is the cost of traversing directly from node ***i*** to node ***k,*** and k is one of the neighbors of node ***i.*** After recomputing the metrics, nodes pass their own distance information to their neighbor nodes again. After a while, all nodes/routers in the network have a consistent routing table to all other nodes.

This protocol does not scale well to large networks due to a number of reasons. One is the so-called count-to-infinity problem. In unfavorable circumstances, it takes up to N iterations to detect the fact that a node is disconnected, where N is the number of nodes in the network [2]. Another problem is the increase of route update overhead with mobility. RIP uses time triggered (periodic, about a 30-s interval) and event-triggered (link changes or router failures) routing updates. Mobility can be expressed as rate of link changes and/or router failures. In a mobile network environment, event-triggered routing updates tend to outnumber time-triggered ones, leading to excessive overhead and inefficient usage of the limited wireless bandwidth.

3.2 Diffie-Hellman Two-Party Agreement (Dh)

This basic protocol, proposed in a landmark paper [3], allows two nodes to build a common key. The principal of this protocol is simple: the two involved nodes, M_1 and M_2, send one another a partial key to be used for the common key computation. M_1 generates a random number $r1$ ($1 \leq r1 \leq p$), and sends α^{r1} to M_2, such that α and p are constants known by each node. On the other hand, M_2 generates a random number r_2, and sends α^{r2} to $M1$. Thereby, each node could compute the common key, which is $\alpha^{r1 \times r2}$ this solution is based on discrete

logarithmic arithmetic, and also relies on the agreement on the parameters α and *p* between the two nodes. Although it is simple and limited to two nodes' common key establishment, this protocol was used to design more sophisticated protocols, as we will see later.

Let we have a prime number 'p' as one whose powers generate all the integers from 1 to p-1, i.e. if 'a' is the primitive root of a prime no 'p', then,

a mod p , a^2 mod p , a^3 mod p, ... a^{p-1} mod p generates all distinct integers from 1 to (p-1) in some permutation.

The steps for Diffie Hellman key exchange algorithm are:

Step 1 : GLOBAL PUBLIC ELEMENTS

Select any prime no : 'q'

Calculate the primitive root of q : 'a' such that a<q

Step 2 : ASYMMETRIC KEY GENERATION BY USER 'A'

Select a random number as the private key X_A where $X_A < q$

Calculate the public key Y_A where $Y_A = a^{X_A} \bmod q$

Step 3 : KEY GENERATION BY USER 'B'

Select a random number as the private key X_B where $X_B < q$

Calculate the public key Y_B where $Y_B = a^{X_B} \bmod q$

Step 4 : Exchange the values of public key between A & B

Step 5 : SYMMETRIC KEY (K) GENERATION BY USER 'A'

$K= Y_B{}^{X_A} \bmod q$

Step 6 : SYMMETRIC KEY (K) GENERATION BY USER 'B'

$K= Y_A{}^{X_B} \bmod q$

It can be easily be proved that the key K generated by this algorithm by both parties are the same.

3.3 General Diffie-Hellman (Gdh)

Steiner *et al.* [4] proposed an n-party generalization of the basic two-party DH protocol (described before). The new protocol consists of *n* rounds, allowing *n* nodes to establish a common key. In the first $n - 1$ rounds contributions are collected from each node. In the first round, M_1 generates $r1$ and computes α^{r1}, which it sends to M_2. In the second step M_2 generates r_2, computes α^{r2} and sends it to M_3, along with α^{r1} and $\alpha^{r1\times r2}$. This latter sends to $M4$ (after making the required computations) the third-round partial factors, i.e., $\alpha^{r1\times r2}$, $\alpha^{r1\times r3}$, $\alpha^{r2\times r3}$, as well as the third-round partial key $\alpha^{r1\times r2\times r3}$. This process continues for each M_i ($i < n$). Upon the $(n - 1)^{th}$ round, the collector node M_n receives the $(n - 1)^{th}$ round partial factors, and the $(n - 1)^{th}$ round partial key, then it generates its random number and computes the final key *K*. In the last round, node M_n sends each M_i the appropriate $(n)^{th}$ round partial factor, i.e.

$$K=\alpha^{(\pi \not{\phi}_{j=1}^{n} r_j)/r_j}$$

Consequently, each node uses its random number to compute the common key *K*.

For example in a network suppose Alice is source and Bob is destination .Alice and Bob want to share a secret key for use in a symmetric cipher, but their only means of communication is insecure. Every piece of information that they exchange is observed by their adversary middle man. How is it possible for Alice and Bob to share a key without making it available to middle man.

International Journal of Advance Research in Science and Engineering
Vol. No.4, Issue 08, August 2015
www.ijarse.com
IJARSE
ISSN 2319 - 8354

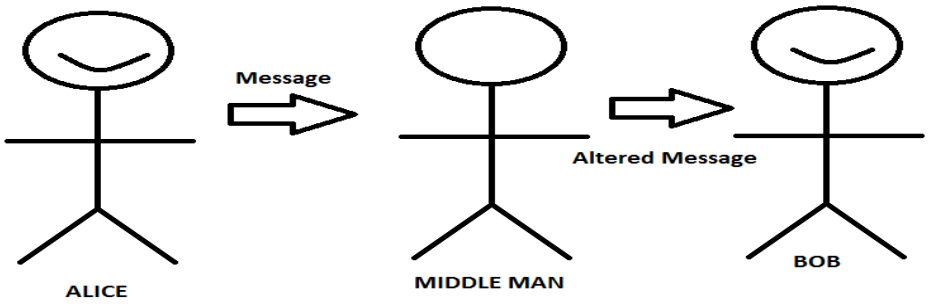


Figure 5.1 : Example of n-party data transfer by key exchange.

This problem is viewed as difficulty of the discrete logarithm problem solved by GDH. The simplest, and original, implementation of the protocol uses the multiplicative group of integers modulo *p*, where *p* is prime and *g* is primitive root mod *p*. Here is an example of the protocol, with non-secret values, and secret values:

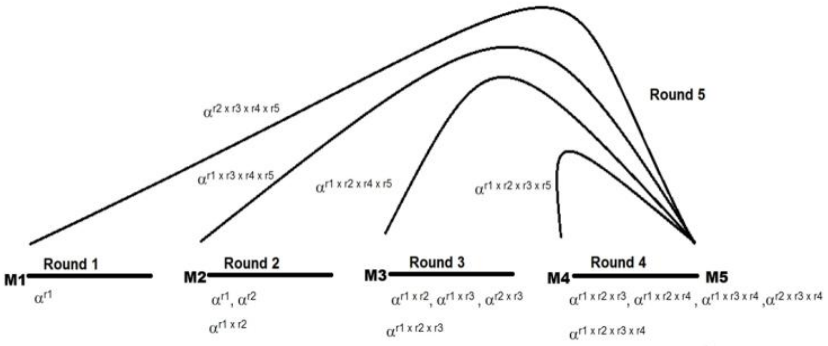
Table 5.1 Key distribution between two parties

ce				**Bob**		
Secret	Public	Calculates	Sends	Calculates	Public	Secret
A	p, g		p,g→			B
A	p, g, A	$g^a \bmod p = A$	A→		p, g	B
A	p, g, A		←B	$g^b \bmod p = B$	p, g, A, B	B
a, **s**	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	b, **s**

1. Alice and Bob agree to use a prime number p=23 and base g=5.
2. Alice chooses a secret integer a=6, then sends Bob A = g^a mod p
   - A = 5^6 mod 23
   - A = 15,625 mod 23
   - A = 8
3. Bob chooses a secret integer b=15, then sends Alice B = g^b mod p
   - B = 5^{15} mod 23
   - B = 30,517,578,125 mod 23
   - B = 19
4. Alice computes s = B^a mod p
   - s = 19^6 mod 23
   - s = 47,045,881 mod 23
   - s = 2

5. Bob computes s = A^b mod p
   - s = 8^{15} mod 23
   - s = 35,184,372,088,832 mod 23
   - s = 2
6. Alice and Bob now share a secret: s = 2. This is because 6*15 is the same as 15*6. So somebody who had known both these private integers might also have calculated s as follows:
   - s = 5^{6*15} mod 23
   - s = 5^{15*6} mod 23
   - s = 5^{90} mod 23
   - s = 807,793,566,946,316,088,741,610,050,849,573,099,185,363,389,551,639,556,884,765,625 mod 23
   - s = 2

Both Alice and Bob have arrived at the same value, because $(g^a)^b$ and $(g^b)^a$ are equal mod p. Note that only a, b and $g^{ab} = g^{ba}$ mod p are kept secret. All the other values – p, g, g^a *mod* p, and g^b *mod* p – are sent in the clear. Once Alice and Bob compute the shared secret they can use it as an encryption key, known only to them, for sending messages across the same open communications channel. Of course, much larger values of a, b, and p would be needed to make this example secure, since it is easy to try all the possible values of g^{ab} mod 23.



Example of General Diffie-Hellman with five Node

PROPOSED WORK In a network when we transfer data from source to destination first of all:

- A network is randomly created.
- Then the shortest distance from source to destination is found with the help of Bellman-Ford algorithm.
- Apply GDH on each node starting from source to destination node on the shortest path calculated.

Step I: In some cases of data transfer there is no intermediate node in between the source and the destination node. Then their is no way that the key shared between the two nodes gets intended and the data transfer is secure.

Step II: If there are some intermediate node in between the source and the destination node i.e. along the path between the source and the destination node Some malicious node intrudes the key and change it and then forwards it to the destined node waiting for the key to be shared, the destined node will not be able to know whether the key coming to be shared is either intruded or not and take it as a key that is not infected from any malicious node (intruder).

Step III: Basically when we transfer data from source to destination. Once source and destination compute the shared secret by applying GDH algorithm, they can use it as an encryption key. This secret key is known only to them, for sending messages across the same open communications channel. Then the source node encrypts the data to be sent with this shared secret key and send it to destination. Only destination node can decrypt this data by using its private key.

Step IV: Then the middle man comes into act. If there is a middle man in between the path of source and destination, then it will try to get the secret key generated by GDH. If he gets the key he can alter the data by using new key and send the wrong data to the destination node. .

Step V: Then the prime target is to detect the malicious node along the whole path and to make the path secure so that data can be transferred securely over the path without any intrusion. Middle man detection is done by the destination node. When the destination node decrypts the data sent by source with its private key, we will check if the Secret key sent by source is same as destination node gets then there will be no middle man (malicious node). But if the is an altered one then there is a middleman in the network.

Step VI: Then we propose a mechanism to avoid the malicious node. We make an authentication pool of nodes in which we assume a thresh hold value of each node. If a node has that value or more than that then it is authenticated otherwise not. We will put all the authenticated nodes into the authenticated pool.

Step VII: So if we found an intruder (middle man) in our shortest path then we will leave that path and try other possible shortest path produced by Bellman-Ford. We will try other paths till we find the new secure path without any intruder or middle man. That new secure shortest path should consist only the authenticated nodes from authenticated pool.

Step VIII: In a case if an authenticated node in the authenticated pool gets infected by the middle man attack. Then we can remove that node from the pool of authenticated nodes.

Step IX: By applying this approach we can find a secure path from source to destination without any middleman for secure data transfer.

IV. IMPEMENTATION AND RESULTS

As we apply over approach to randomly created network, we will get these results:

```
Enter The number of nodes
10
        0       1       2       3       4       5       6       7       8       9
0       0       94      9       28      61      97      4       7       17      40
1       94      0       32      78      29      8       49      60      48      42
2       9       32      0       3       62      52      70      5       70      42
3       28      78      3       0       10      40      47      33      46      1
4       61      29      62      10      0       11      90      56      90      26
5       97      8       52      40      11      0       38      58      44      27
6       4       49      70      47      90      38      0       2       96      60
7       7       60      5       33      56      58      2       0       37      56
8       17      48      70      46      90      44      96      37      0       86
9       40      42      42      1       26      27      60      56      86      0

Enter The Node with which u wanna communicate
8
0 -> 8 ->
Selecting Prime Numbers P and G
Secure Key: 506
**********************************
Checking the authenticity of network
Sending Data 506
Sending Encrypted Data though the network
Destination node received the message
Decrypting data
There is a middle man
########################################
########################################
```

As we enter the number of nodes a network is generated randomly with the number of nodes entered. This simulator will pick node 0 as a source node to resolve the network complexity. After that when we give the destination node. It has found out the shortest path between source and destination given by using bellman-Ford algorithm. Then it picks two random prime numbers p and g (p and g are prime numbers because it is difficult to have factors of prime numbers, so it will be difficult to hack). By using p and g the value of GDH (Secure secret key) has calculated. In results that is 506. Then source node that is 0 is sending data by encrypting it with secure secret key (506).

```
Press 1 to continue
Press 0 to exit
1
Let there be a middle man in the Network
2
2
2
**************************************
Checking the authenticity of network
Sending Data 77
Sending Encrypted Data though the network
Destination node received the message
Decrypting data
There is a middle man
##########################################
##########################################
Press 1 to continue
Press 0 to exit
1
Authenticated Diffie Hellman
Enter The Node with which u wanna communicate
8
0 -> 8 ->
Authenticated Nodes are:
0        8        9        5        7
Authenticating Route
All nodes are authenticated
Selecting Prime Numbers P and G
Secure Key: 84
```

V. CONCLUSION & FUTURE WORK

In the randomly created networks when we transfer data from source to destination first of all we find out the shortest path. It will minimize the cost of data transfer and resolve the complexity of the network. The shortest path has find out by Bellman ford algorithm. In some cases we have only two nodes source and destination, then the data transfer is secure. But if there exists more than two nodes then it will be difficult to transfer the data from source to destination in a safe manner so that no one can access the data.Also the research work is restricted to limited number of nodes in the future work we can add number of nodes.Also the efficiency and accuracy of transfer of data can also be enhanced and overhead of time and complexity can also be overcome.

REFERENCES

[1] Bashir Yahya, Jalel Ben-Othman, "Achieving host mobility using DNS dynamic updating protocol", In the Proceedings of LCN 2008, Pages: 634-638, October 14-17, 2008; Montreal, Canada

[2] Bongsoo Kim, Younghwan Choi, Kwansoo Jung, Hochoong Cho, Fucai Yu, and Sang-Ha Kim "A Dynamic Single-Hop Clustering Mechanism Adapted to Overlay Multicast in Mobile Ad hoc Networks" Dept. of Computer Engineering, Chungnam National University, 220 Gung-dong, Yuseog-gu, Daejon, Korea, 369-764, {bskim, yhchoi, ksjung, chohc99, yufc}gcclab.cnu.ac.kr and shkimgcnu.ac.kr

[3] R. Lin and M. Gerla, ”Adaptive clustering for mobile wireless networks”, IEEE Journal on Selected Areas in Communications, 15(7), pp. 1265-1275, Sepember 1997

[4] D. J. Baker, A. Ephremides, and J. A. Flynn, ”The design and simulation of a mobile radio network with distributed control”, IEEE Journal on Selected Areas in Communications, SAC-2(l), pp. 226-237, January 1984

[5] Kershaw, “Some extensions of W. Gautschi’s inequalities for the Gamma function,” Math. of Computation, vol. 41, pp. 607–611, Oct. 1983.