International Journal of Engineering & Technology, 7 (3) (2018) 1453-1456

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i3.14219
Research paper

Routing protocol based key management schemes in manet: a survey

Shibu K.R [1] *, SujiPramila R [2]

[1] Research scholar,Department of CSE,Noorul Islam University,Nagercoil,Tamilnadu
[2] Associate Professor,Department of CSE,Noorul Islam University,Nagercoil,Tamilnadu
**Corresponding author E-mail:shibukarakkattu@gmail.com*

Abstract

Mobile ad-hoc networks (MANETS) are infrastructure less networks and the topology of this network is always changing. The nodes can enter and leave the network at any time. These networks require a high security in communication, as its application demands so. Effective key management is the only technique, which can implement to secure the nodes in communication. In adhocnetworks there is no central controller or router such as in wired network. This will make this network more vulnerable to attack. The intruders can easily enter the net-work and can manipulate the contents easily. Traditional key management schemes will not fit for this type of networks. This article, discuss various key management schemes based on routing protocols in mobile ad-hoc network (MANET). It also analyses them in terms of the security and applicability.

Keywords:*Manet; Key Management; CA; Certificate Chaining;*

1. Introduction

Mobile ad-hoc networks are temporary networks with no centralized access point, such as base station, access points etc. The only entity in this network is nodes which are free to move randomly. The nodes can communicate each other within the range, also the node themselves act as routers instead of routers in wired networks. This makes network more vulnerable to attacks. So a proper key management scheme is a major requirement, to achieve integrity, security, robustness etc. If a source node wants to send some messages to the destination, then it has to take the help of intermediate nodes to transfer the message. This intermediate nodes help to relaying the message from source to destination. These kinds of networks are useful in various situations such as emergency rescue operations, military battle field. For e.g. during fire attacks, the building as well as the network will collapse, then it is very difficult to recover also it will take time and cost. In such case the application of ad hoc network arises. This will help the people who are engaged in this rescue operation to setup networks temporarily to manage the team activities. There are various other applications also, which are outside the scope of this paper. In order to achieve security in mobile ad-hoc networks different key management schemes are there. The evaluation of this key management is based on different parameters. One of the important parameter among them is complexity. This paper evaluates three key management schemes with three different routing protocols and their complexity levels.

A description on key management schemes based on three different protocols is discussed in the paper. In section, II elaborates on Diffie-Hellman key exchange, which is the base for all key management schemes [1]. In sections III, IV and V, an effort has been taken to study three different routing based key management schemes, their merits and demerits. Section VI concludes the paper.

2. Diffiehellman key exchange

Diffie Hellman is one of the oldest and basic key management schemes. It is a symmetric key management scheme[1]. If a source node wants to communicate with a destination node, they should share some secret key to ensure the data security. It will assure that an eavesdropper (Eve) can't overhear the communicated message.

In this algorithm eve is considered as a passive attacker, who can only see the message and can't modify it. In this method source and destination can exchange a secret key without Eve can learn it and this key is then used for further communication. The algorithm is as follows

Algorithm 1. Diffie Hellman

```
  1)  Randomly select two large numbers, one prime 'S', and 'G' a primitive root of P
  2)  Users pick private values 'a' and 'b'
  3)  Compute values
$X = G^a \bmod S$
$Y = G^b \bmod S$
  4)  Values 'X' and 'Y' are exchanged
  5)  Compute shared, private key
$K_a = Y^a \bmod S$
$K_b = X^b \bmod S$
  6)  Algebraically it can be shown that Ka = Kb
```

By the above method a secret key agreement is developed between the source and the receiver nodes. The main advantage of this scheme is the fact that the key is not transmitted over the network, it is generated by the nodes themselves. So if an intruder wants the key he has to generate it using the values exchanges by the two nodes. And the probability of finding the key by a third node is very small. This algorithm will work fine for communication between two nodes but have drawbacks such as, no user authentication and computational complexity. Even though this algorithm contributed too much for the later key generation schemes, this method will not work for ad-hoc networks and for larger networks.

3. Key generation using routing metadata

In this method, the routing metadata is used for key generation [2]. It is based on the concept of random key generation using a data existing in the system itself [3]. The routing information details are identified as ideal for key generation and the data can be easily accessible by all the entity in the network. For this method, it has to seek help from routing protocols such as Dynamic Source Routing (DSR) or Adhoc On demand Distance Vector (AODV).

In the paper discussed here DSR is used due to the following features [4]. In DSR there are two phases; a) Route discovery and b) Route maintenance. The nodes in the network will maintain an ordered list of the nodes through which the packet moves from source to destination and this information is added in to the header. This information is having some inherent randomness and it is used as the source for the randomness.

3.1. Dynamic source routing

DSR is a well known protocol designed for wireless ad-hoc network. In mobile adhoc networks since the topology is changing frequently, the traditional protocols for wired networks will not work. So protocol such as DSR is used. Consider an example, in fig1 suppose node N1 want to establish a route with N5.

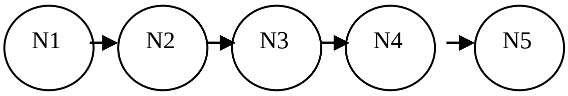

Fig. 1:. Routing in DSR.

The node1 will initiate a route discovery, by transmitting a route request (RR) packet. The RR packet is placed on the source route position of the packet header. The RR packet will be foreword to N2, N2 will foreword that packet to next and finally it will reach N5 which is the destination node. Whenever the packet moves from one node to another the header will be updated with the address of the intermediate node through which the RR packet flows. When the packets reach a node, it will check the destination node address in the current nodes data store, if a match is found it will stop forwarding the packet, and that node will respond with a Route Reply (RRP) which contains the valid path from source to destination. Otherwise, the RR packet will be forwarded till it reaches the destination node.

Now the destination node will give the route reply. DSR is a reactive routing protocol i.e. it will find out a route only when a request comes. Every node in the network has a route cache, which contain valid routes, whenever a node receives a new path that will be updated in the route cache. Route cache will store the source IP, destination IP and route address.

3.2. System overview

In the system the properties of the dynamic source routing algorithm is utilized for key generation. In DSR after route discovery phase the nodes cache will have the details of path from source to the target node. The length of the path will depend on various parameters and it can vary for each route request.

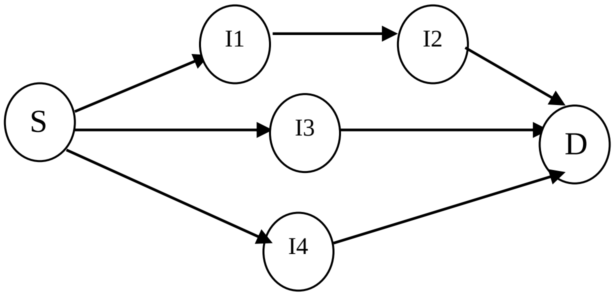

Fig.2: Communication between S and D.

To explain the details consider a communication scenario as shown in fig.2. S and Dare the source and destination nodes and let intermediate node I2 is the route reply sender. Then after Dynamic source routing, the node cache of all the nodes involved in the route will have the entries as given in the table I.

Table 1: Route Table

Route ID	Partial Route	Full Route
S-D-RID-I2	S-RID-I2	S-I1-I2-D

Once routing information is obtained then key generation is done using the entries available in the route table. The first step is to agree on some common randomness between the source and destination. It is performed by information reconciliation. In this KERMAN algorithm is used for information reconciliation [5]. The KERMAN algorithm is given below.

```
Algorithm 2. KERMAN
1) If Node 'S' wants to communicate with 'D' then 'S' send an availability check message to 'D'.
2) If 'D' is available then reply is send to 'S'
3) 'S' identify all the partial route entries in its route table where 'D' is present.
4) Then these selected data RIDs are forwarded to 'D'
5) 'D' check its own route table and identify the mismatches and inform that to 'S'.
6) Now 'S' and 'D' have some common secret between them.
7) It is then used for key generation.
```

The problem with this method is that when RIDs are exchanged in clear there is a possibility that the eaves dropper can overhear the details. Thus it may give a chance for eaves dropper to know the identities of many routes that is included in the RIDs.

Table 2: Groups when RID is Send

Group	Type	Source	Destination	RRP Sender
	1	S	D	X
1	2	S	X	D
	3	X	S	D
	4	S	X	Y
2	5	X	S	Y
	6	X	Y	D
3	7	X	Y	Z

To be precise this may expose five nodes i.e. three from RID (Source IP, Destination IP, Route Replay IP) and 'S' and 'D' nodes IP. These possibilities are tabulated in the table II. In this table the group 1 entries shows the data leakage of one unknown node in addition to source and destination. In group 2 details of two nodes are revealed to the adversary and in the third group all the three are unknown nodes and all together the adversary is getting the details of five nodes.

To overcome this draw back the key generation is done after privacy amplification. This step includes the methods to increase the randomness of the secret shared random data. Here the source and destination nodes agree on a partitioning algorithm and based on that algorithm they create a set of RIDs. Then these sets are further subdivided randomly into small subsets. These subsets are assigned with binary values which are random enough to generate the keys for communication.

This method has the main advantage that the source of randomness is readily available in the system itself and also the computation complexity is less. Above these advantages the method has the drawback that the key generated depends on the route length, so it can lead to packet overhead. It also lacks the ability to identify active attackers.

4. Mobility based key management technique

In this scheme key is generated based on the mobility of the nodes in network [6]. This type of key management scheme work well for cluster based networks. Cluster Based Routing Protocol (CBRP) is used here.

4.1. Cluster based routing protocol

Clustering is the process of splitting networks into smaller networks in distributed manner. Each of the clusters would have a cluster head, which is the controller of that particular group. This node will retain the information about all the nodes in the particular cluster. The head nodes can communicate with other head through the sinks, where sinks are base stations. Mobility based key management schemes are invoked when a node moves from one cluster to another.

In this method the pair wise key is generated with the involvement of the cluster head and the sink. The figure 3 shows a cluster head based network. There will be a numerous number of clusters in given distance.

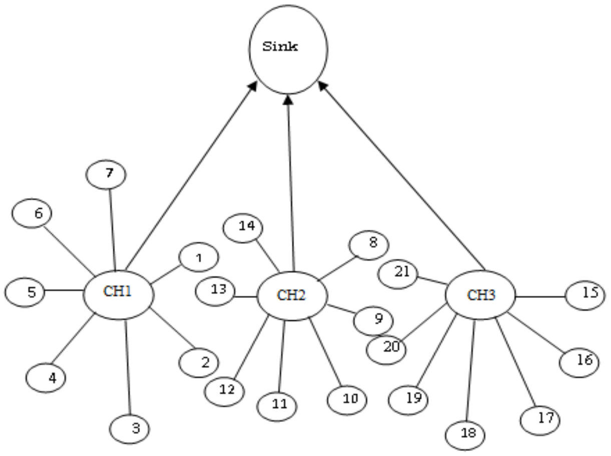

Fig. 3:. Cluster Based Manet.

Here the main design issue is the selection of the cluster head. The selection of cluster head is based on three criteria 1) Energy resource 2) Range of Communication 3) Processing capability. Once the above criteria are met, then that node is chosen as a candidate for the selection. Then this candidate node will broadcast a head request packet to all the nodes in the cluster to get selected as the head. All the nodes will respond with a reply CHREP. If the number of reply reaches a threshold T_h then that node will be selected as cluster head (CH), which is shown in equation 1.

$$\text{COUNT (CHREP)} >= T_h \tag{1}$$

Once the CH is selected it will send a message to all the nodes regarding the cluster head <cluster ID, node ID>. Also it will send the information about its cluster to the sink. The sink will give a cluster key $K_{ch}$to the cluster and preloaded in the nodes.

4.2. Key generation

Whenever a node say, $N_k$wants to communicate with other node say $N_j$and if $N_{k,}N_j$ are members of the same cluster the cluster head will authenticate them for communication. Then the node N_k will send an encrypted message by a key K_i and send to the other. .The node can receive this message only if they are having the decrypting key, which is already preloaded.

$$\text{Encrypt (CH } \{K_i\}) \tag{2}$$

The above method works on the assumption that nodes which are not closer to cluster head should not communicate each other to achieve security. Whenever a node moves from one cluster to another it will send CHREQ to the sink. Then an authentication code is generated by the sink to identify the new node. Before adding to the cluster the new cluster head will verify with the sink, which is shown in equation 3.

$$\text{CHREQ } (N_i) \rightarrow \text{Sink} \tag{3}$$

Only after the successful verification, the sink will generate the key pair as shown in equation 4.

$$\text{GEN (CREQ, } K_i) \tag{4}$$

The reply from the sink will consist of CH_{id}, pair wise key etc. The algorithm of mobility based key management is given below.

Algorithm 3. Mobility Based Key Management

```
1) $N_k$ be the node want to communicate with $N_j$
2) $N_k$has to generate a request CHREQ
3) $CH_i$ will check both of the nodes belong to same cluster.
4) If $N_{k,}N_j$are members of the same cluster they can communicate.
5) Else $CH_i$ forward CHREQ to Sink
6) Sink has to Authenticate ($N_i$) based  on old($CH_i$)
7) If (auth($N_i$) = = true)
1.1      Verify using MAC
1.2      If (MAC = = TRUE)
1.2.1    Generates the key and transmit it to the cluster head
1.3      Else exit(Unsuccessful)
2.       $N_k$and $N_j$ can communicate now.
```

The key generation in this method requires some data structures. The sink and cluster head will maintain two tables. The table in the sink consist of Cluster member Id, pair wise key, lifetime etc. Similarly the table in the cluster head will contain cluster name, cluster head id, key life etc. The number of entries in this table will vary based on implementation. Whenever a change occurs in the cluster it should be updated in these tables.

This scheme is very effective for security in mobile node based systems. But attacker inside the cluster can easily access the data and can become a threat. This key management scheme fit only for adhoc network that are cluster based.

5. Composite key management scheme

This scheme combines the features of Public key infrastructure with distributed certifying authority and key chaining [7]. These approaches are not effective at all when they are considered alone. In this approach the positive features of both techniques are merged together to achieve a high level security. These combined schemes make them adaptable to dynamic changes in the network. CA plays an important role to make the communication more trustworthy [9]. The CA should act like a normal node at the same time it should be trustable. The CA can be single node or a group of nodes, which issues digital certificate to other nodes who are communicating each other. If there are multiple CA nodes then the load of key generation and distribution is evenly distributed. Here if one technique fails then the network can utilize the other, this make the composite key management scheme more reliable.

5.1. Frame work for key management

This section discusses how to generate a framework with the existing techniques. The types of nodes in this key management scheme are 1) CA Node 2) Nodes in Certificate Chaining 3) Client

Nodes. The role of CA node is to create partial signature, renew the certificate and also to maintain a log of certificates issued. All the nodes involved in certificate chaining can issue certificates to its nearby nodes. Each client node is calculating its own confidence value based on common criteria.
There are numerous methods for initialising a node as CA. In the modern environment the functionality of CA is distributed across different nodes. For e.g. S is the set of nodes which act as CA’s

$CA = \{M_1, M_2, M_{3,}M_n\}$

Here ‘n’ is the number of certifying authority. So if a node wants to communicate, they need only a certificate from the subset of nodes from the above set. That means only a few CA [8] nodes are needed for certification process. The proper selection of CA contributes much to this. A certification graph is used to denote the trust relationship among the adjacent nodes. A sample graph is shown in the figure 4. The nodes represent the key pair and the value marked along the edge shows the confidence level, which can vary from 0 – 1. If the value is less than 0.5 then it means no trust.

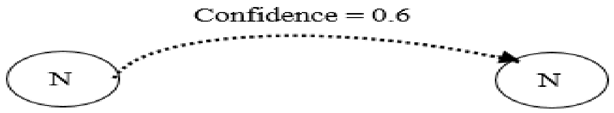

Fig. 4: Key Chain Graph.

To communicate from a source to target all the chains existing between them are considered and the confidence value(C value) of each one is calculated. The confidence value of a single chain is obtained by multiplying the value of all edges in that path. As an example consider the figure 5. The effective C value can be calculated by the equation below;

$$C\ value= K_1 * K_2 * K_3. \qquad (5)$$

Each and every node in the network is a probable adversary ‘p’, so the probability of a chain being safe is expressed as a function of the route distance ‘l’ with safety factor ‘S_f’;

$$S_f = (1 – p)^{(l-1)} \qquad (6)$$

Now the final confidence value is calculated by multiplying the values of equation (5) and (6). After computing these values for each chain the decision to permit communication is granted if and only if the C value is above a preset threshold value.

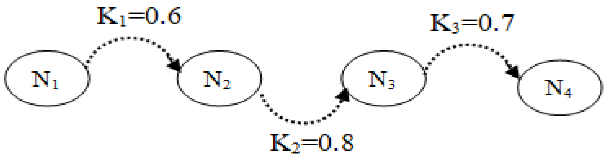

Fig. 5: Sample Certificate Chain.

If any node among this set is an eavesdropper then the entire system will crash, here the selection of this nodes plays a key role for successful communication. To avoid such failures a Trusted Third Party based frame work can be introduced or in some systems the entire nodes in the network will be treated as CA’s. In such system the performance of the network will depend on the number of CAs. Another factor affecting the C value is the route length, as it increases higher probability to include malicious nodes. So measures should be taken to avoid lengthy key chains.

Table 3: Performance of Key Management Schemes

	Parameters		
Key Management Schemes	Security	Network Size	Robust
Routing Metadata	High	Applicable to large networks	Yes
Mobility Based	Low	Limited	None
Composite Key	High	Depends on Cluster size	Yes

6. Conclusion

The table III gives a comparative study of the above discussed techniques in terms of applicability, security and network size and node mobility. The performance of all the three methods relies on the respective routing protocols. The main constrain of all the scheme is route length, as it increases the possibility of eavesdropping also increases. Neither of the schemes considers the possibilities of active attackers and does not included any measures to overcome such challenges.

References

[1] Diffie, Whitfield, and Martin Hellman. "New directions in cryptography." *IEEE transactions on Information Theory* 22, no. 6 (1976): 644-654.https://doi.org/10.1109/TIT.1976.1055638.

[2] Khalili-Shoja, Mohammad Reza, George TraianAmariucai, Shuangqing Wei, and Jing Deng. "Secret common randomness from routing metadata in ad hoc networks." *IEEE Transactions on Information Forensics and Security* 11, no. 8 (2016): 1674-1684https://doi.org/10.1109/TIFS.2016.2550424.

[3] Park, Stephen K., and Keith W. Miller. "Random number generators: good ones are hard to find." *Communications of the ACM* 31, no. 10 (1988): 1192-1201.https://doi.org/10.1145/63039.63042.

[4] Johnson, David B., David A. Maltz, and Josh Broch. "DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks." *Ad hoc networking* five (2001): 139-172.

[5] Shoja, Mohammad Reza Khalili, George TraianAmariucai, Shuangqing Wei, and Jing Deng. "KERMAN: A key establishment algorithm based on harvesting randomness in MANETs." *arXiv preprint arXiv: 1504.03744* (2015).

[6] Eschenauer, Laurent, and Virgil D. Gligor. "A key-management scheme for distributed sensor networks." In *Proceedings of the ninth ACM Conference on Computer and Communications Security*, pp. 41-47. ACM, 2002.https://doi.org/10.1145/586110.586117.

[7] Capkun, Srdjan, LeventeButtyán, and J-P. Hubaux. "Self-organized public-key management for mobile ad hoc networks." *IEEE Transactions on mobile computing* two, no. 1 (2003): 52-64.

[8] Yi, Seung, and Robin Kravets. "Composite Key Management for Ad Hoc Networks." In *MobiQuitous*, vol. 4, pp. 52-61. 2004.

[9] Dahshan, Hisham, and James irvine. "A trust based threshold cryptography key management for mobile ad hoc networks." in *vehicular technology conference fall (vtc 2009-fall), 2009 ieee 70th*, pp. 1-5. ieee, 2009.