Volume 6, Issue 12, December 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Secured Communication in Multicast Network using Diffie-Hellman Algorithm

Amandeep Chhabra, Pragya

Department of Computer Science & Engineering, U.I.E.T,K.UK

Haryana, India

Abstract: Mobile ad-hoc networks (MANET) is a collection of mobile nodes communicates with each others, but has no fixed links like wireless infrastructure networks. Each node work as router and each router is responsible for dynamically discovering other nodes who can directly communicate with each other. Multicasting which is an efficient means to support key applications of mobile ad hoc networks (MANET) such as tele-conferencing .Due to the dynamic network topology of an ad hoc network, the network changes frequently and unpredictably, so the security of multicast routing becomes more challenging than the traditional networks. The key exchange technique is identical to the standard broadcast version of the Diffie-Hellman public-key algorithm. However, from an implementation point of view, nodes within a multicast group are treated in a binary fashion, where a shared secret key is generated for a pair of nodes at a time. Once the shared secret key is calculated by the pair, nodes within the pair are treated as a single entity by a node that is to be joined. This process is recursively performed until all the nodes in the multicast group attain a common shared secret key. In this paper, we narrate how any user in the multicast group can compose the group keys and use the group Diffie- Hellman key-exchange protocol (GDH) to securely multicast data from the multicast source to the rest of the multicast group in wireless ad hoc networks.

Keywords: Manet's, Multicast Security, Diffie-Hellman Algorithm

I. INTRODUCTION

A MANET[3] is an autonomous system of mobile nodes. The routing, power management, bandwidth management, radio interface, and security are hot topics in MANET research. Although in this paper we only focus on the security issues in MANET. The overall goal of the security solutions for MANET is to provide security services including authentication, confidentiality, integrity, and availability to the mobile users. In order to achieve these goals, the security solution should provide complete safe spanning the whole protocol stack. We can categories MANET security in 5 layers, such as *Application layer, Transport layer, Network layer, Link layer,* and *Physical layer.* However, we only focus on the network layer, which is related to security issues, to protect the routing and forwarding protocols. From the security design perspective, the MANETs have no clear line of defense. Unlike wired networks that have edicated routers, each mobile node in an ad hoc network may works as router and forward packets for other nodes. Most of the previous work has focused mainly on providing preventive schemes to protect the routing protocol in a MANET[1]. Most of these schemes are based on key management or encryption techniques to prevent unauthorized nodes from joining the network. In general, the main pitfall of these approaches is that they introduce a heavy traffic load to exchange and verify keys, which is very expensive in terms of the bandwidth-constraint for MANET nodes with limited battery and limited computational capabilities. Multicast plays an important role in MANET. Many ad hoc network applications need the nodes to work as a group to carry out a given job. Multicasting is the communication of data packets to more than one node sharing same multicasting address. The senders and receivers form the multicast group or other group. There could be more than one sender in a multicast group, so it is group-oriented computing. In wired networks, some well established routing protocols can provide efficient multicast, but when it comes to MANETs, these protocols may fail due to some unique characteristics of MANETs. As a result, multicast routing has become a research focus recently, and security algorithm in MANET have been proposed.

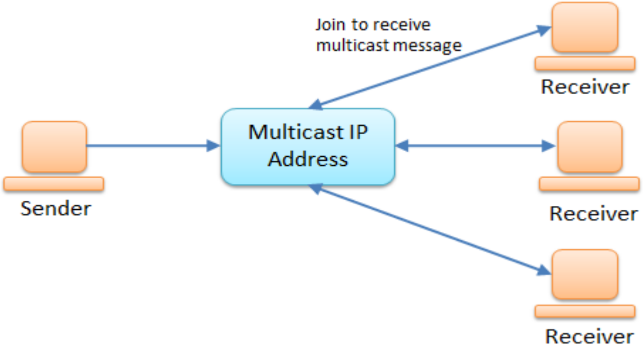


Fig-1 Multicasting

In the above figure the sender is sending the message through single multicast IP address to all the receiver to which it wants to send the mesaage and the same concept is implemented in the paper by considering total 6 nodes.

II. MULTICAST SECURITY

The Domain of multicast networking and security related issues is a wide technical subject. Within the limitations allowed, we discuss few relevant technical issues and performance tradeoffs to observe when applying security and key management techniques in support of multicast networking[2]. First we consider the application of existing and proposed security techniques for multicast networking, including key distribution, dynamic key management, and reliability issues. Throughout in the paper we summarize security policy and algorithm to secure Multicast networks.

III. PROPERTIES OF MULTICAST

The definition of the host group model provides a summary of the key properties of multicast. A host group is a set of network nodes sharing a common multicast address, all receiving data packets addressed to this multicast address by source that may or may not be members of the same group and have no knowledge of the group membership. This definition highlights the three main properties of multicast:

- All the member nodes receive all packets sent to the address: Multicast routing delivers all packets sent to the multicast address to all members of the multicast group.
- Open group membership: Multicasting provide an open group model and allows group membership to be transparent to the source.
- Open access to send packets to the group: Any host can send data to the multicast address, and it will be delivered to the multicast group without regard for the source of these packets.

IV. REQUIREMENTS OF KEY MANAGEMENT IN MULTICAST

The main job of key management in multicast is generating,distributing and updating group key for its member.Group key is held by all members and used for encryption and decryption datagram. The messages are protected by encryption using the chosen the group key. Only those who know the group key are able to recover the original message. The group key need to be refreshed when the group have membership change. The main approaches to key management are[4] :

1. Key Predistribution:

As the name suggests key pre distribution involves distributing keys to all interested parties before the start of communication. once deployed, there is no mechanism to include new members in the group or to change the key.

2. Key transport:

In key transport system ,one of the communicating entities generates keys and transports them to the other members. It assumes that a shared key is already exists among the participating members. This prior shared key is used to encrypt a new key and is transmitted to all corresponding nodes. Only those nodes which have the prior shared key can decrypt it. This is called Key encrypting method (KEK)[4].

3. Key Arbitration:

Key arbitration schemes use a central arbitrator to create and distribute keys among all participants. Hence, they are a class of key transport schemes. There is a difference in distribution of public keys which belong to a public knowledge, and private (secret) keys which are shared by multiple entities. Private keys can be distributed through a pre-established secure channel or an open channel. Public keys are usually distributed through certificates. A certificate binds a public key with an entity.

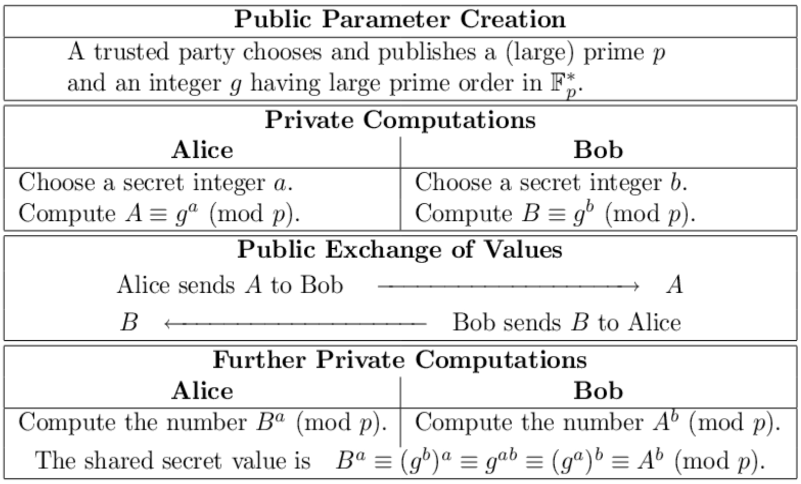
4. Key agreement:

Most of key agreement schemes are based on asymmetric key algorithms. Key agreement is used when it is necessary for several parties to agree upon a secret key and its exchanges, used in later communications. In case of group key agreement, each participant contributes a part to the secret key. This involves high computational complexity. The most popular key agreement scheme is Diffie-Hellman exchange which we will discuss in the paper.

V. DIFFIE HELLMAN ALGORITHM

The Diffie-Hellman[5] key exchange protocol is a cryptographic protocol that was developed by Whitfield Diffie and Martin Hellman in 1976.

The Diffie-Hellman key exchange algorithm solves the following dilemma. 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 Eve. How is it possible for Alice and Bob to share a key without making it available to Eve? At first glance it appears that Alice and Bob face an impossible task. It was a brilliant insight of Diffie-Hellman that the difficulty of the discrete algorithm problem provides a possible solution. The explanation of the Diffie-Hellman Algorithm is given above in the figure-2.



Public Parameter Creation	
A trusted party chooses and publishes a (large) prime p and an integer g having large prime order in $\mathbb{F}_p^*$.	

Private Computations	
Alice	**Bob**
Choose a secret integer a. Compute $A \equiv g^a \pmod{p}$.	Choose a secret integer b. Compute $B \equiv g^b \pmod{p}$.

Public Exchange of Values

Alice sends A to Bob $\xrightarrow{\qquad\qquad}$ A

B $\xleftarrow{\qquad\qquad}$ Bob sends B to Alice

Further Private Computations	
Alice	**Bob**
Compute the number $B^a \pmod{p}$.	Compute the number $A^b \pmod{p}$.

The shared secret value is $B^a \equiv (g^b)^a \equiv g^{ab} \equiv (g^a)^b \equiv A^b \pmod{p}$.

Fig-2 Diffie-Hellman Algorithm

VI. RELATED WORK

A lot of research has been done in the field of providing security to Multicast ad hoc networks[7] but all these research done does not fully fulfill the essential security in Manets. Many Group diffie hellman protocols aim to provides a session key among the multicast group members for a scenario in which the membership is static and known in advance. However these protocols are not well-suited for a scenario in which members join and leave the multicast group at a relatively high rate. As described by[8] formalization of group diffie hellman key exchange and the adversary capabilities is done. In the formalization, the nodes do not deviate from the protocol, the adversary is not a node and the adversary capabilities are modeled by various queries. These queries provide the adversary a capability to initialize a multicast group via set-up queries, add nodes to multicast group via join-queries and remove nodes from multicast group via remove-queries. Another simple and efficient region based group key management scheme is proposed, simply called SERGK[9], for MANETs. The basic idea of SERGK is that a physical multicast tree is created in MANETs for efficiency. Group members now turns acting as group coordinator to compute and distribute intermediate key materials to group members. The keying materials are delivered through the tree links. The coordinator is also responsible for maintaining the connection of the multicast group. All group members can calculate the group key locally in a distributed manner.

The 2-party Diffie-Hellman exchange was first proposed in 1976, there have been efforts to extend its simplicity and elegance to a group setting. The Authors of [10] discuss a useful DH-based multi-party key-generating technique for large static groups, called Group-DH. The principal of this protocol is simple: the two involved nodes, $M1$ and $M2$, send one another a partial key to be used for the common key computation. $M1$ generates a random number p), and sends a$r1$ to $M2$, such that a and p are constants known by each node. On the other hand, $M2$ generates a random number $r2$, and sends a$r2$ to $M1$. Thereby, each node could compute the common key, which is This solution is based on discrete logarithmic arithmetic, and also relies on the agreement on the parameters a 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.

VII. PROPOSED GROUP DIFFIE-HELLMAN TECHNIQUE

Step 1: A member acts as a group controller and forms a two-party group with the remaining group members. Each group individually generates a DH-style key using DH technique[6].

Step 2: The group controller generates (n-1) public keys by raising the exponent of g with the product of (n-2) shared keys at a time and sends to the corresponding group members. On receiving, each member raises the exponent With its own shared key and generates the group key.

Proposed m-party key distribution protocol (Improved group DH):

Let $P1$,2,…,Pi …,Pm−1 ,Pm be the group members and let Pi (1≤i≤m) acts as a group controller.

Initially “ ” itself forms a two-party group with each of the remaining group members, and produces (n-1) two-party groups.

Pi selects a private key xi and generates a public key :

$Xi =gxi\ mod\ n$

and broad cast to the remaining group members. Also each group member , where j ≠ i also assumes a private key and generates a public key as

$Xj =gxj\ mod\ n$

Where xj is the private key of pj and 1 ≤ j ≤ n, j ≠ i .

where xj is the private key of Pj and 1≤j≤ n, j≠i.

Each Pj then transmits Xj to the group controller, . After exchanging the public keys, each member similar to the basic DH generates a unique shared key, Ki with group controller as

$Ki=Xixj\ mod\ n=gxixjmod\ n$

Similarly, Pi generates the same shared key, using

$Ki=Xjxi\ mod\ n=gxjximod\ n$

It Actually generates (n-1) shared key Ki’s for $1 \le j \le m$ and $j \neq i$ for *(n-1)* parties respectively.

To produce a single group key first group controllers computes the following $Zl's$,encrypt with $Kl's$ respectively and send to $Pl\ 's$ respectively, for 1< *l* <m ,*l≠ i.*

After receiving each Pl decrypts with their key and computes the group key k as follows.:

$Zl= \pi Ki\ j\neq lmod\ n,$

$where$ $1\le l\le m$,$\neq i,j\neq l$. Each party in the group then generates the group key ,k ,as follows:

$P1$Generates, $K=Z1\times K1\ mod\ n= \pi Kij\neq imod\ n$

P2Generates, $K=Z2\times K2\ mod\ n= \pi Kij\neq imod\ n$

Pm-1 Generates, $K=Zm-1\times Km-1\ mod\ n=\pi\ Kij\neq imodn$

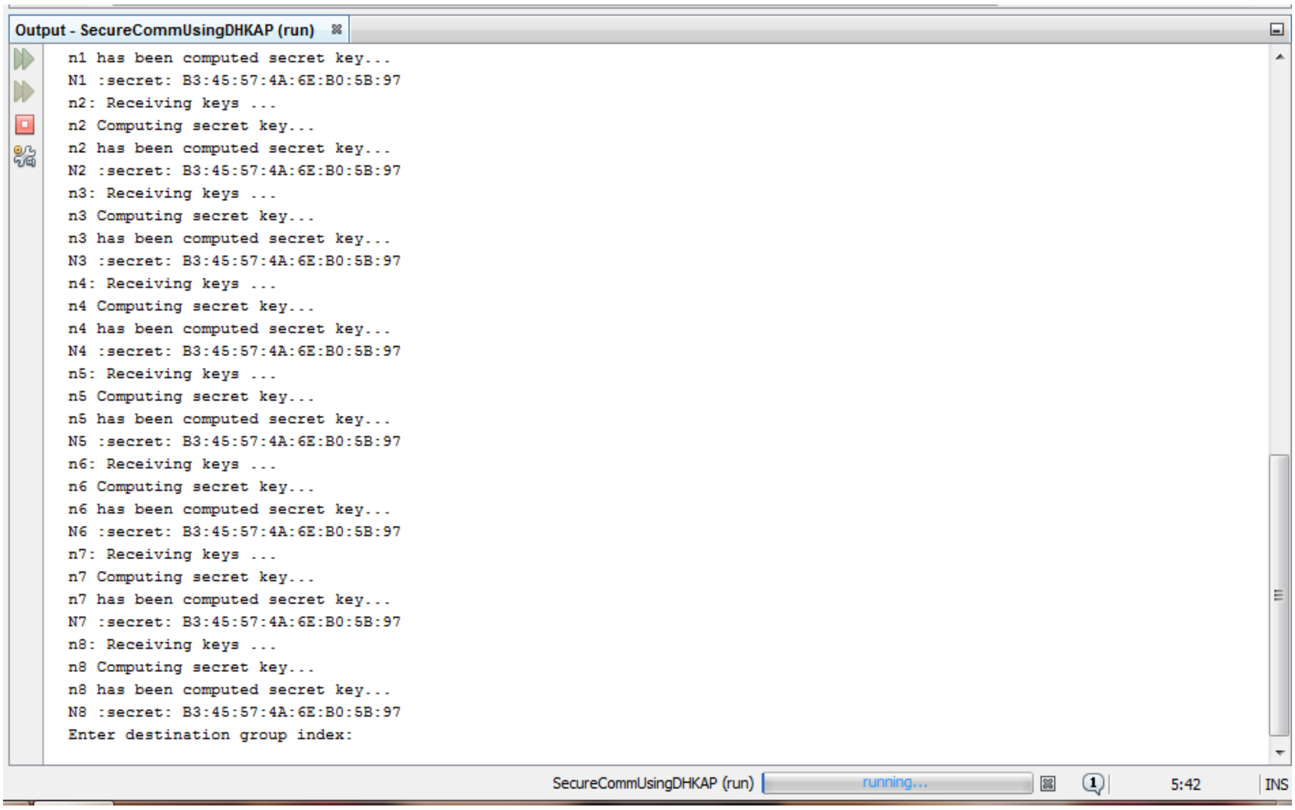
Pm Generates, $K=Zm\times Km\ mod\ n= \pi Kij\neq imod\ n$

Since the group controller knows all the two party shared keys it also generates the group key using

$K= K\pi ij\neq i\ mod\ n$

VIII. IMPLMENTATION RESULTS

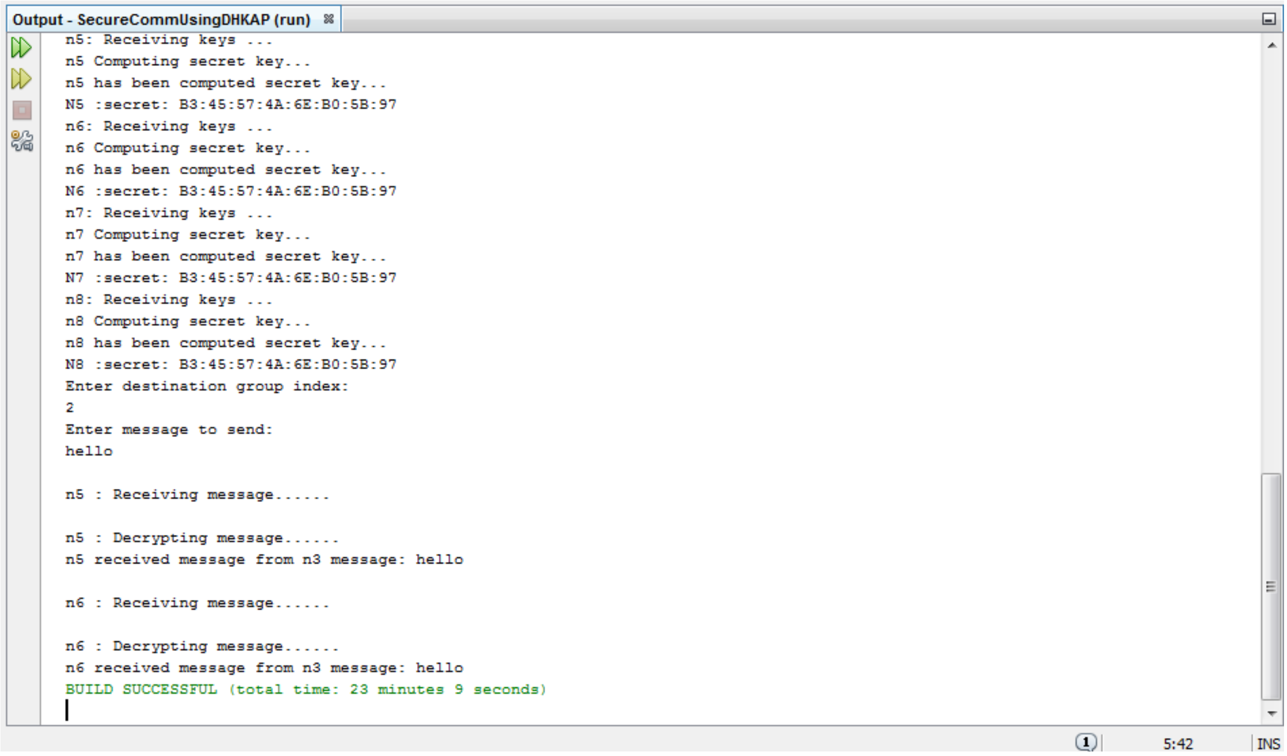
There are many types of attacks possible on adhoc networks as discuss in[11].Cryptography is one of the most common and reliable means to ensure security. It can be applied to any communication network[4].In cryptography the original information to be sent from one person to another is called plain text. This plain text can be converted into cipher text by the process of encryption. An authentic receiver can decrypt/decode the cipher text back into plain text by the process of decryption. The process of encryption and decryption are governed by keys, which are small amount of information used by the cryptographic algorithms. When the key needs to be kept secret to ensure the security of the system, it is called secret key. The secure administration of cryptographic keys is called key management[4].The four main goals of cryptography are confidentiality, integrity, authentication and non-repudation. There are two main cryptographic algorithms : symmetric key algorithms ,which use the same key for encryption and decryption and asymmetric key algorithms, which uses two different keys for encryption and decryption[4].The aim is to implement Key Agreement approach using Diffie hellman Algorithm for Multicasting protocol in which all the nodes in the network agree on the same key as said by the key agreement approach. For two nodes the algorithm has been implemented in[6].Our work has been extended to n-party communication in which n parties or nodes can participate in communication.


Output - SecureCommUsingDHKAP (run)
SecureCommUsingDHKAP (run)
running...
5:42
INS

```
n1 has been computed secret key...
N1 :secret: B3:45:57:4A:6E:B0:5B:97
n2: Receiving keys ...
n2 Computing secret key...
n2 has been computed secret key...
N2 :secret: B3:45:57:4A:6E:B0:5B:97
n3: Receiving keys ...
n3 Computing secret key...
n3 has been computed secret key...
N3 :secret: B3:45:57:4A:6E:B0:5B:97
n4: Receiving keys ...
n4 Computing secret key...
n4 has been computed secret key...
N4 :secret: B3:45:57:4A:6E:B0:5B:97
n5: Receiving keys ...
n5 Computing secret key...
n5 has been computed secret key...
N5 :secret: B3:45:57:4A:6E:B0:5B:97
n6: Receiving keys ...
n6 Computing secret key...
n6 has been computed secret key...
N6 :secret: B3:45:57:4A:6E:B0:5B:97
n7: Receiving keys ...
n7 Computing secret key...
n7 has been computed secret key...
N7 :secret: B3:45:57:4A:6E:B0:5B:97
n8: Receiving keys ...
n8 Computing secret key...
n8 has been computed secret key...
N8 :secret: B3:45:57:4A:6E:B0:5B:97
Enter destination group index:
```

Fig-3 Secret key generation

As the figure 3 shows secret key is being generated for certain number of nodes which wants to communicate in the network in a secured way using n-party Diffie-Hellman Algorithm.



```
n5: Receiving keys ...
n5 Computing secret key...
n5 has been computed secret key...
N5 :secret: B3:45:57:4A:6E:B0:5B:97
n6: Receiving keys ...
n6 Computing secret key...
n6 has been computed secret key...
N6 :secret: B3:45:57:4A:6E:B0:5B:97
n7: Receiving keys ...
n7 Computing secret key...
n7 has been computed secret key...
N7 :secret: B3:45:57:4A:6E:B0:5B:97
n8: Receiving keys ...
n8 Computing secret key...
n8 has been computed secret key...
N8 :secret: B3:45:57:4A:6E:B0:5B:97
Enter destination group index:
2
Enter message to send:
hello

n5 : Receiving message......

n5 : Decrypting message......
n5 received message from n3 message: hello

n6 : Receiving message......

n6 : Decrypting message......
n6 received message from n3 message: hello
BUILD SUCCESSFUL (total time: 23 minutes 9 seconds)
```

Fig- 4 Message delievery to nodes

Figure 4 above shows message delievery to different nodes in the network which already shared secret key using Diffie-Hellman technique.

IX. CONCLUSION

In the paper various issues related to Manet's and Multicasting are presented. Various Key Management issues have been explained. Then we present Diffie-Hellman algorithm technique for 2-party communication. Further, Group Diffie Hellman(GDH) Algorithm is implemented statically with 4 groups and 2 nodes in each with a total of 8 nodes and can be increased upto n-nodes. The Message communicating between the nodes is communicating securely as all the nodes uses secret key to encrypt and decrypt message.

REFERENCES

[1] Rashid Hafeez Khokhar, Md Asri Ngadi, Satria Mandala," *A Review of Current Routing Attacks in Mobile Ad Hoc Networks"*, in International Journal of Computer Science and Security, volume (2) issue (3).

[2] Peter S. Kruus, Joseph P. Macker," *TECHNIQUES AND ISSUES IN MULTICAST SECURITY"* Naval Research Laboratory Washington, DC 20375.

[3] Nishu Garg, R.P.Mahapatra," *MANET Security Issues*" in IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.8, August 2009.

[4] C.Siva ram murthy , B.S.Manoj ," *Adhoc wireless networks architecture and protocols*".

[5] Sunita, Neeraj Goyat, Annu Malik," *Review of Diffie – Hellman key Exchange*" in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 7,July 2013.

[6] Vankamamidi S.naresh, Nistala V.E.S Murthy," *Diffie-Hellman Technique Extended to Efficient and Simpler Group Key Distribution Protocol"* in International Journal of Computer Applications (0975 – 8887) Volume 4– No.11, August 2010.

[7] Wenjing Lou, Wei Liu , Yuguang Fang," *SPREAD: Enhancing Data Confidentiality in Mobile Ad Hoc Networks"* in ieee infocom 2004.

[8] Emmanuel Bresson, Olivier Chevassut & David Pointcheval ," *Provably Authenticated group diffie hellman key exchange – the dynamic case".*

[9] Elizabeth M.Royer , Charles E Perkins," *Multicast operation of the adhoc on demand distance vector routing protocol".*

[10] Vankamamidi S. Naresh , Nistala V.E.S. Murthy," *Diffie-Hellman Technique Extended to Efficient and Simpler Group Key Distribution Protocol* " in International journal of computer Applications august 2010.

[11] Ajay Jangra, Nitin Goel, Priyanka& Komal Bhatia*," Security Aspects in Mobile Ad Hoc Networks"*