DOI: 10.21917/ijct.2021.0346

A METHOD FOR OBTAINING SECURE AND EFFICIENT GROUP KEY OVER WIRELESS AD-HOC NETWORKS BASED ON A VIRTUAL SUBNET MODEL

P.J.A. Alphonse and Y. Venkatramana Reddy

Department of Master of Computer Application, National Institute of Technology, Triuchirappalli, India

Abstract

Today, mobile devices such as PDAs and notebooks etc are became most important in human life. A MANET is a mobile ad-hoc network which supports all types of communication such as unicast, multicast and "many-to-many" transmission. Until recently, efficient methods were not devised to generate group key for such groups. Generally, communication in wireless networks is, the risk of sensitive information being intercepted by unintended recipients is a real concern. So, MANETs require efficient and secure group communication. With the model, the groups are established as the forming of group keys. Our results show that this approach can completely satisfy the today's needs of both security and efficiency for group communication.

Keywords:

Diffie-Hellman Protocol, Secret Key Distribution, Virtual Subnet Model, Group Signature

1. INTRODUCTION

The tremendous amount of data exchange and sharing can be supported by the development in networking technology. The computers on different networks can communicate by using Virtual local area network (VLAN) [1]. A VLAN operates like a general LAN, but the devices my not have a physical connection in the same network. When they need cross-regional communication, VLAN technology can group them together, and exchange the information as simply as in a LAN. A mobile ad - hoc network (MANET) [2] is formed by grouping the wireless mobile nodes have special characteristics such as wireless communication and networking. They can freely form communication groups as necessary. In general, a particular application or interest may need to establish the corresponding community. Generally, any number of communication groups may exist in the same MANET. So, the network is a collection of communication groups as in a VLAN. The groups may propagate packets simultaneously. In broadcast transmission data packets can be received by all groups or nodes, including those that do not need or should not receive them. The sensitive information is explored publicly. Many methods have resolved security issues in this networks, such as mesh networks [3], sensor networks [4] [5] and inter cluster key management [6], to the best of our knowledge group construction and secure group communication have not drawn much attention.

In this article we design a virtual subnet model to construct group communication in a MANET which achieves security and efficient communication. There is an initiation and authentication phase to select and authenticate the members, a construction and key agreement phase to construct groups and share the secure key in a group. In addition, we provide revocation, tracing and transmission mechanisms that cooperate to achieve the efficient and secure virtual subnet behaviour and communication.

The remainder of this article is organized as follows. In section two we describe VLAN, section three we reviewed the [10]. Section four describes our protocol for constructing secure and efficient group key agreement in Wireless Ad-Hoc Networks. Finally, we make concluding remarks and identify some possible directions for future study.

2. VIRTUAL LOCAL AREA NETWORK

Broadcast and multicast packets in the traditional switched LAN are always forwarded to all devices even the nodes that do not require them. To solve this problem, the IEEE 802.1Q [7] standard was developed to divide a large network into smaller ones so that broadcast and multicast traffic do not require more bandwidth than necessary. The IEEE 802.1Q is a logical collection of network devices which include how the frames are relayed to destinations. The major points are:

- Any frame belonging to a VLAN has a VLAN-tag which is used to find the nodes belongs to that VLAN.
- The routing information of all groups (nodes) is stored in filtering database (FDB).

VLAN-aware switches can make filtering or forwarding decisions for packets, communicate with other switches and routers within the network.

The frame is checked for errors when it arrives at the VLAN-aware switch. Error free frames are associated with a VID, and error frames are dropped. The frame is rejected, if the ingress filter (source port filter) is set to enable and the incoming port is not a member of the same VLAN.

The frame which is accepted is entered the forwarding process to be relayed to other ports; meanwhile, the switch studies the information about the frame, such as VID and data, and uses it to update the FDB if required. Frames can be forwarded using the MAC address and VID of the frame indices which are stored in the FDB.

3. REVIEW ON SECURED GROUP COMMUNICATION [10] FRAMEWORK

Key exchange is the foundation of secure group communication. In the original literature, the two-party Diffie-Hellman key exchange protocol was proposed in 1976 [8]. There are two well-known system parameters in the protocol: q is a prime number, and α is a primitive root that is less than q. If A and B need to share a secure key, they create random private values a and b, respectively. Then they generate their public values by the parameters q and α. A's public value is $x_a = \alpha^a \bmod q$ and B's public value is $x_b = \alpha^b \bmod q$. Finally, they exchange their public values; A and B share a common secret key by the following equations:

A computes

$$(x_b)^a \bmod p = (\alpha^b \bmod p)^a \bmod p = (\alpha^b)^a \bmod p \qquad (1)$$

and Bob computes

$$(x_a)^b \bmod p = (\alpha^a \bmod p)^b \bmod p = (\alpha^a)^b \bmod p \qquad (2)$$

In [9], the Diffie-Hellman key exchange protocol has extended to *n*-party setting, and the security is as robust as the original two-party protocol.

The GDH.3 in [9] comprises four steps, assuming all participants $U = \{u_1, u_2, \dots, u_n\}$. Agree to share a secure key. In the first step each individual u_i, provides its contribution x_i to u_{i+1}, by upflow, where $1 \leq i \leq n-2$. In the second step u_{n-1}, processes the final upflow message to obtain $\alpha^{a_1 a_2 \dots a_{n-1}} \bmod q$ and broadcasts this value to all other participants. In the third step, participant u_i $(i \neq n)$ receives the value $\alpha^{a_1 a_2 \dots a_{n-1}} \bmod q$, factors out its own exponent, then forwards the result to u_n. In the last step, u_n receives each value from the previous stage, raises its power a_n to every one of them, and broadcasts the resulting *n*-1 value to the rest of the group. At this stage, each u_i has a value with the form $\alpha^{\prod\left\{a_j | j \in \{1,\dots,n\} \text{ and } j \neq i\right\}}$ and can compute the common group key by raising this value to the power of a_i. Finding the key $\alpha^{\prod\left\{a_j | j \in \{1,\dots,n\} \text{ and } j \neq i\right\}}$ for each u_i is very difficult. Because, *n*-1 keys are broadcasted to the members of the group.

In 2007, the Huang et al. [10] extended GDH.3 to accommodate a MANET environment and enhances the security to prevent eavesdropping and tampering with the information between group key agreement procedures. To describe the procedure, the following notations are used in their procedure.

n - Number of participants in a virtual subnet

i, *j* - Index of virtual subnet members between 1and *n*

M_i - i^{th} virtual subnet member between 1 and *n*

X_i - i^{th} private value

Y_i - i^{th} public value

First, each member node of the virtual subnet generates a random private value X_i. In accordance with the ascending order of the virtual subnet member list, each node contributes its public value $Y_i(a^{X_i})$ to gather in M_{n-1} by unicast protocol.

M_1 computes R_1 then forwards R_1 to M_2, M_2 solves a^{Xi} and raises its private value X_2 to a^{X_1} then forwards to M_3 and so on. Consequently, node M_{n-1} will receive R_{n-2} as Eq.(4) in Fig.1.

In Eq.(5), node M_{n-1} raises its X_{n-1} and transmits the result R_{n-1} to other nodes in the same virtual subnet by multicast. Then in the first step $M_i(i \neq n)$ receives the value R_{n-1} and gets $a^{x_1, x_2, \dots, x_{n-1}}$. Each node factors out its own private value X_i, then forwards the result R_i to node M_n as Eq.(6). In the same manner node M_n gets R_i of node M_i, raises its X_n, and then sends the result R_n to node M_i.

Finally, each node M_i has the value

$$\alpha^{\prod_j^n x_j \left(j \subset [1,n] \text{ and } j \neq i\right)} \qquad (3)$$

In this framework we have two possibilities to send the partial key $\alpha^{\prod\left\{a_j | j \in \{1,\dots,n\} \text{ and } j \neq i\right\}}$ for each u_i. In the first method, by using unicast communication, user u_n can send the partial key $\alpha^{\prod\left\{a_j | j \in \{1,\dots,n\} \text{ and } j \neq i\right\}}$ for each u_i. In the second method, user u_n can send the partial key $\alpha^{\prod\left\{a_j | j \in \{1,\dots,n\} \text{ and } j \neq i\right\}}$ for each u_i by encrypting with shared key between u_i and u_n.

After it raises its own private value X_i and mod *q* operation, the secure group key will be generated confidentially.

$$M_1 \xrightarrow[]{\text{Computes } R_1=h(ID_1)\oplus a^{x_1}} M_2 \xrightarrow[\text{Solves } a^{x_1}= R_1\oplus h(ID_1)]{R_2=h(ID_2)\oplus a^{x_1 x_2}} M_3 \rightarrow \cdots \rightarrow M_{n-1} \qquad (4)$$

$$M_{n-1} \xrightarrow{R_{n-1}=h(ID_{n-1})\oplus a^{x_1 x_2 \dots x_{n-1}}} M_1 \qquad (5)$$

$$M_{i(i\neq n)} \xrightarrow{R_i'=h(ID_j)\oplus a_j^{\pi x_j^{n-1}} \; where\, i, j \subset [1, n-1] and\ j \neq i} M_n \qquad (6)$$

$$M_{n-1} \xrightarrow{R_n=h(ID_j)\oplus a_j^{\pi x_j^{n}} \left(j \subset [1,n] and\ j \neq i\right)} M_1 \qquad (7)$$

Fig.1. Group Key Agreement Procedure

Their model conceals the private value under the hash function and exclusive-or operations, and even parameters *q* and *a* are only known by those nodes in the same virtual subnet instead of all nodes. This can guarantee that the group key is generated confidentially, and effectively prevents eavesdropping and tampering in the group key exchange procedures.

The main limitations of their protocol are first, in the group initiation stage the agent node should be trusted otherwise attackers could be included in the group. Second, the virtual subnet information (list of virtual subnet members and other information) is transferred to each member of that virtual subnet the size of that information is depend on the size of the virtual subnet. So the creation of virtual subnet with very more number of members is impractical. Third, sending the partial keys from user u_n to the remaining users is not efficient. Fourth, a node may quit or join a group over time, therefore, a sophisticated group regeneration method must be developed to facilitate this MANET feature. Our model is designed to solve the above limitations and provides secure communication.

4. OUR CONTRIBUTION

In our framework we design group key for *m* entities virtual subnet. We select *m* entities from *n* entities need to communicate. Here *n* is the number of entities available in the world. In the same way our framework can create any number of virtual subnets with *n* entities. Every entity can store only one private key and generate group keys of the virtual subnets it belongs to by using its private key and partial group key of the concerned virtual subnet. This framework provides both confidentiality and authentication.

This section can be divided into six phases namely Initiation and Authentication phase which explains the selection and authentication of *m* members from *n* members. If we design groups by selecting and authenticating members, that virtual subnets are completely protected from Man-In-the-Middle attacks. Construction and key agreement phase which explains the design of virtual subnets and partial keys of the members of that virtual subnets. Communication which explains the generation of group key from partial keys of individual entities. Maintenance which explains cache table (routing table) update procedure in our framework. Revocation which explains the prevention of dishonest entities from the virtual subnet and departure of one entity from the virtual subnet by himself. Finally tracing which explains discovery of the sender of group message.

4.1 INITIATION AND ATHENTICATION PHASE

The member who wants to create a group based on some topic or interest, broadcast group formation request over the wireless network. Those who are interested (m members) to join that group (grouping can be formed in any manner) can send the willingness message by specifying their public key to the initiator. Initiator will form a group by selecting and authenticating m members, we authenticate m members from the available n members by using the below algorithm.

4.1.1 Digital Signature-based Approach:

The initiator of the virtual subnet can authenticate the m members by using the public keys of selected m members. In our algorithm group size is m which includes the initiator of the virtual subnet. The algorithm to authenticate m-1 members is given below:

Algorithm: Member Authentication Algorithm

```
The initiator would select message $M$, send to all.
For ($i = 1; i \le n; i$++)
  $w = h(M)$, $0 \le w \le q-1$
  Choose random integer $k_i$, $x_i$ such that $1 \le k_i \le q-1$ and $gcd(k_i, q-1) = 1$
  $s_i = \alpha^{k_i} \bmod q$
  Compute $k_i^{-1} \bmod (q-1)$
  $s_{ii} = k_i^{-1}(m - x_i s_i) \bmod (q-1)$
  Transfer ($s_i$, $s_{ii}$) to the initiator
$j = 1$
while ($j \le M-1$)
  $v_i = a^w \bmod q$
  $v_{ii} = (y_i)^{s_i}(s_i)^{s_{ii}} \bmod q$
  If $v_i = v_{ii}$ then select $i$ and $j$++
```

The signature is valid if $v_i = v_{ii}$. Let us demonstrate that this is so. Assume that this equality is true for member i. Then we have,

$$a^m \bmod q = (y_i)^{s_i}(s_i)^{s_{ii}} \bmod q \tag{8}$$

Assume $v_i = v_{ii}$

$$a^m \bmod q = a^{x_i s_i} a^{ks_2} \bmod q \tag{9}$$

Substituting for y_i and s_i

$$a^{m-x_i s_i} \bmod q = a^{ks_{ii}} \bmod q \tag{10}$$

Rearranging terms

$$m\text{-}x_i s_i \equiv ks_{ii} \bmod (q\text{-}1) \tag{11}$$

Property of primitive roots

$$m\text{-}x_i s_i \equiv kk^{-1}(m\text{-}x_i s_i) \bmod (q\text{-}1) \quad \text{Substituting for } s_{ii} \tag{12}$$

4.2 CONSTRUCTION AND KEY AGREEMENT PHASE

Let x_i the private key of the member has priority i, α is a primitive root and p is the prime number.

In our protocol, we have given priority to the set of users $U = [u_1, u_2, \ldots, u_n\}$. Here u_i means the user who has the priority i. the main purpose of the priority is that for each phase only two members can compute the partial keys. Specifically, on the first phase who have priority n-1 and n-2 can compute the partial keys the remaining members cannot perform any operations. In general, on the ith phase the members who have priority n-i and n-(i+1) can compute partial keys. Similarly on (n-1)th phase the member who have priority n-(n-1) that means, u_1 can compute the group key for the virtual subnet.

We explain how to devise group key for five member's virtual subnet by using our protocol. We are listing the virtual subnet members in ascending order of their priority in the table specified Fig.2. The partial keys specified in the ith column are computed by the member who has priority i. To reduce the size of the table, we eliminated mod p operations for all partial keys. For example in the second phase the member u_3 could broadcast two partial keys to the group, the first partial key is $\alpha^{x_4^{x3}} \bmod p$, the second partial key is $\alpha^{x_5^{x3}} \bmod p$. When the generation of the partial keys is over, they can broadcast to the group.

Each phase in the table below is divided into two rows. The first row specifies the keys broadcast to the group, and the last row specifies the partial keys computed by the members. When computation of the partial keys is over, each partial key is broadcast to the group independently. The partial keys generated in the ith phase can broadcast in the (i+1)th phase. These keys received by the all members in the group. So, we merged the columns of the first row of each phase.

Now we explain the procedure to create group key for five members $U = [u_1, u_2, u_3, u_4, u_5\}$by using the table Fig.2 shown below. In this protocol the variable used to store partial key can have three indexes, like $ps_{x,y,z}$ the first index x specifies the priority of the member who generated the key, the second index y specifies the phase number in which the key is generated, the last index z specifies the number. Specifically, it is either it is the first key or second key or the last key generated by the user u_x in yth phase.

Let $ps_{5,1,0} = \alpha^{x5} \bmod p$; $ps_{4,1,0} = \alpha^{x4} \bmod p$. These partial signatures of the members u_5 and u_4 are the public keys the members u_5 and u_4 respectively.

Phase 1: The members who have priority n-1+1 and n-1 that is the members u_5 and u_4 are created their partial keys in the previous phase (or before this phase) can broadcast to the group is specified in the first row of the phase. The member who has priority n-1, n-(1+1) that is the members u_4, u_3 can take the partial keys has the priority greater than n-1, n-(1+1) respectively. They obtain their partial keys by raising their received partial keys to the power of their own private key. Specifically the member u_4 can take the partial key which has priority 5 and obtain his partial key by raising the received partial key to the power of his own private key. Similarly, u_3 can obtain his partial keys by raising the received partial keys (the partial keys having priority 4 or 5) to the power of their own private key.

Phase *i*: The members who have priority n-i+1 and n-i that is the members u_{n-i+1} and u_{n-i} are created their partial keys in the previous phase (or before this phase) can broadcast to the group is specified in the first row of this phase. The member who has priority n-1, n-(1+1) that is the members u_{n-i}, $u_{n-(i+1)}$ can take the partial keys has the priority greater than n-1, n-(i+1) respectively. They obtain their partial keys by raising their received partial keys to the power of their own private key. Specifically the member u_{n-i} can take the partial key which has priority n-i+1 and obtain his partial key by raising the received partial key to the power of his own private key. Similarly, $u_{n-(i+1)}$ can obtain his partial keys by raising the received partial keys (the partial keys having priority n-i, n-(i+1) to the power of his own private key.

Phase *n*-1 (last phase):The members who have priority n-(n-1)+1 and n-(n-1) that is the members u_2 and u_1 are created their partial keys in the previous phase (or before this phase) can broadcast to the group is specified in the first row of this phase. The member u_2 can broadcast his partial key to the group. The members who have priority n-(n-1), n-(1+1) that is the members who have priority 1 or 0 (in our list no member is having priority 0), the member $u_{n-(n-1)} = u_1$ can take the partial key has the priority greater than 1 and obtains his group key by raising the received partial key to the power of his own private key. Specifically the member u_1 can take the partial key which has priority 2 and obtain the partial key by rising the received partial key to the power of his own private key. This partial key becomes the group key generated by the member u_1.

In the last phase only the member u_2 can create the partial key and send it to the group. The member u_1 can create the group key by taking the partial key received from group and raising this to the power of his own private key.

The first member can take some message M, encrypt it with group key and send it along with the n-1 partial signatures created in the (n-2)th phase and M to the group. All the members in the group (except u_1) try to decrypt the encrypted message by using the partial keys received from u_1. Each member succeeds with only one partial key, that partial key becomes the partial key of that member that partial key is stored in pk_i for user u_i.

u_1	u_2	u_3	u_4	u_5	Phase
α^{x_4}, α^{x_5}					1
		$\alpha^{x_4x_3}$, $\alpha^{x_5x_3}$	$\alpha^{x_5x_4}$		
$\alpha^{x_4x_3}$, $\alpha^{x_5x_3}$, $\alpha^{x_5x_4}$					2
	$\alpha^{x_4x_3x_2}$, $\alpha^{x_5x_3x_2}$, $\alpha^{x_5x_4x_2}$	$\alpha^{x_5x_4x_3}$			
$\alpha^{x_4x_3x_2}$, $\alpha^{x_5x_3x_2}$, $\alpha^{x_5x_4x_2}$, $\alpha^{x_5x_4x_3}$					3
$\alpha^{x_4x_3x_2x_1}$, $\alpha^{x_5x_3x_2x_1}$, $\alpha^{x_5x_4x_2x_1}$, $\alpha^{x_5x_4x_3x_1}$	$\alpha^{x_5x_4x_3}$				
$\alpha^{x_5x_4x_3x_2}$					4
$\alpha^{x_5x_4x_3x_2x_1}$					

Fig.2. Group signature generation for five members' virtual subnet

4.3 GENERALIZED PROTOCOL

In this protocol the variable used to store partial key can have three indexes, like $ps_{x,y,z}$ the first index x specifies the priority of the member who generated the key, the second index y specifies the phase number in which the key is generated, the last index z specifies the number. Specifically, it is either it is the first key or second key or the last key generated by the user u_x in yth phase.

```
GeneralizedDH ()
$ps_{n,1,0} = \alpha^{x_n} \bmod p$;
$ps_{n-1,1,0} = \alpha^{x_{n-1}} \bmod p$;
For $i$=1 to $n$-1 do
  For $u_{n-i} \in U$
    $ps_{n-i,i+1,0} = (ps_{n-i,i+1,0})^{x_{n-i}} \bmod p$
    For $u_{n-(i+1)} \in U$
      For $j$ = 0 to $i$ do
          $ps_{n-(i+1),i+1,j} = (ps_{n-i+(j \text{ div } i),i,\, j \bmod i})^{x_{n-(i+1)}} \bmod p$
        For $u_1 \in U$
          Select message $M$
          $C = E_{ps_{1,n-1,0}}(M)$
          Transfer the $n$-1 partial keys generated in ($n$-2)th phase,
          $C$ and $M$ to the members.
          For $u_i \in U$; $1<i\leq n$
            For $j$ = 0 to $n$-2 do
              $pk_i = (ps_{1,n-1,j})^{x_i} \bmod p$
                If $M=D_{pk_i}(C)$ then
                  $ps_{1,n-1,j}$ is the partial key for the member $u_i$
                End
            End
          End
        End
      End
  End
End
```

The time complexity of our algorithm, is linear time, the time complexity of the present existing protocols is either exponential or polynomial. This protocol is most efficient compared with all existing protocols. This protocol uses n-1 iterations to generate partial keys for all members. Each iteration only two members are involve to generate temporary partial keys. The data transferred between any two members and the group is very less. After the completion of n-2 iterations (phases) the first member can calculate partial keys of the remaining members. The broadcast message in (n-1)*th* iteration is the partial key of the first member, by raising this partial key to the power of his own private key, can generate his group key.

After the completion of n-1 iterations, the first member can take some message M, encrypr it by using his group key, The encrypted message, original message and the n-1 partial keys generated in (n-2)th iteration are broadcasted to the group. Every member they try to decrypt the message by raising the partial keys received from the relay node to the power of their own private keys and compare with original message. By using this comparison they can find out their partial keys. Our protocol is most secured because the private keys cannot be transferred between members and between member and relay node.

4.3 COMMUNICATION

Each node can initiate a communication to the virtual subnet to which it belongs. The packets include a VSID (virtual subnet ID) to trigger a group communication.

When any member *i* wants to send the message *M* to his group members, he will generate group signature by using his partial group signature

$$S = (pk_i)^{x_i} \mod q \tag{13}$$

He can Encrypt the message $C = E_s[M, VSID, pk_i]$.

The header part of the message pk_i will be used for tracing purpose, VSID which is used to transfer the message to the members of that virtual subnet by using cache table (routing table). Every member of the virtual subnet will generate group key by using his private key and his partial group key of that group and decrypt the message. For example member *j* will generate the group key and decrypt the message "*C*" by using the below procedure

$$S = (pk_i)^{x_j} \mod q$$
$$m = D_s(c) \tag{14}$$

The node will accept the packet that belongs to the same virtual subnet and reply with an acknowledgement (ACK) packet to the source node. If any node of the virtual subnet does not reply with an ACK packet in the normal way, the source node is aware that the node lost its hop nodes or the path has been changed. At this moment the source node initiates router discovery using a unicast algorithm to find the routing path to the node. It then sends the CERQ to the hop nodes in the new path and resends the lost packet to the node

Non-group entities (attackers) are not able to decrypt the message even he knows the partial signatures of all members of that group, because group key is a combination of partial group key and private key of the concerned individual group entity, he do not know the private key of at least one member.

4.4 MAINTENANCE MECHANISM

The forwarding cache table in this model is equivalent to the filter in a VLAN and it is created by using any dynamic routing algorithm. Neighbour relationship will not be static, so the forwarding cache table needs to insert and delete related information at the proper time to ensure that the records in the cache table are up to date.

In order to insert a VSID into the forwarding cache table, each node periodically advertises a CREQ packet that contains its node ID and VSID. When a node is inserted the cache tables of each entity could be updated by using any dynamic routing algorithm, this can maintain the shortest path between every pair of entities. A VSID will be deleted from the forwarding cache table when the node neither receives the CREQ packet nor forwards the packets to the virtual subnet of the VSID representation in a period of time. The deletion is necessary to reduce redundant communications when nodes move away from their original positions or exceed the radio range so that its original neighbour nodes do not need to forward the packet. At the same time, the new neighbour nodes must record the VSID in their forwarding cache tables and the forwarding tables of the remaining nodes will be updated.

4.5 REVOCATION

In our article we are dealing with two types of revocations. The first one is some member/ members of the group is not able to continue in that group, He/they can exit from the group. We need to remove the group accessing rights of that member/members. The second one is, how to prevent dishonest entities from the group.

Case 1*:* If any member *i*, wants to exit from the group, he can compute the group key $s = (pk_i)^{a_i} \bmod q$ and broadcast his un-willingness packet.

$[E_S, (pk_i), VSID]$ The message is transferred to the members of the virtual subnet over the network.

Non-group members are not able to send the above message on behalf of another group member as they don't know the private key of any member of that group. When the un-willingness message is received by the members of that group, the members of that group can do the following operations.

Each member *j* will calculate the group key $s{=}(pk_j)^{x_j} \bmod q$. He will decode partial key available in the received packet.

The received partial key of the member *i* becomes the group key of the remaining members. Because, $s = a^{a_1\, a_2 \ldots a_{i\text{-}1},\, a_{i+1} \ldots a_m}$ mod q. The remaining *m*-1 members will update their partial key by using the following loop

For *i* = 1 to *m*-1;

$$pk_i = (s)^{\frac{1}{x_i}} \bmod q \tag{15}$$

Case 2*:* if the group entities want to prevent dishonest entities, this operation can be done by the authorised entity, he must be a member of the group. The authorised entity can request partial group signature of the dishonest entity *i*. The authorised entity *j* can compute the group signature $s{=}(pk_j)^{a_j} \bmod q$ and broadcast the unauthorised entity *i*'s exit message $[E_S(pk_i)]$ over the network. The dishonest entity's partial signature will become the group signature for the remaining group entities. All entities, including authorised entity will update its partial group signatures by sing the following loop.

For *k* = 1 to *m*-1;

$$pk_k = (pk_i)^{\frac{1}{x_i}} \bmod q \tag{16}$$

The dishonest entities, who are not interested to give their partial signatures to the authorised entity can be revocated from the group by using the construction and Key agreement phase specified in section 4.2.

4.6 TRACING

Some part of the message specifies the partial key of the sender of the message. By using that part of the message we can trace the sender of the message. Partial key will act like membership certificate for the member and it will also authenticate the member. Tracing is very easy in our framework.

4.7 SCALABLE

In our framework virtual subnets are scalable. Every member in the communication system can become a member of any number of virtual subnets. Internal storage of the members of the virtual subnet does not depend on the size of the subnet. The size of the message is constant and does not depend on the virtual subnet size. The responsibility of the initiator is to generate group and its group key only. We are not using any central authority for any purpose. But if we want to prevent dishonest entities, prevention authority can be given to any member. All members are independent.

4.7.1 Join:

When the generation of a virtual subnet is over, if any one wants to join a group, he will broadcast the willingness message to a member of that virtual subnet. A member of that virtual subnet can do the following operations.

Let a_k, x^{a_k} be the private and public keys of the new member k who wants to join the group respectively.

The member i of the virtual subnet will authenticate the new entity by using authentication algorithm (which is used to protect the group from unauthorized entities) specified in Fig.1. The entity i can generate group key $s = (pk_i)^{x_i}$ mod q, create the message $M = E_x a_k [S]$, send it to member k over network. The entity k can decrypt the message $s = D_{a_k}[M]$ becomes the partial signature of the new member that means $pk_k = (pk_i)^{x_i} \bmod q$. He can generate the new group signature $s^1 = (s)^{a_k}$, create message $M = E_s[s^1]$ and broadcast [M, $VSID$] over the network.

All entities belongs to that virtual subnet, including authorised entity will update its partial key by using the following loop.

For i = 1 to m

$$S = (pk_i)^{x_i} \bmod q,\ s^1 = D_s[M] \tag{17}$$

$$pk_i = (s^1)^{\frac{1}{x_i}} \bmod q \tag{18}$$

5. COMPARISON

To generate final partial key, our protocol requires n-1 phases, each phase requires one time slot, the two members broadcast the partial keys to the group plus one time slot is used to transfer the keys and data from the member u_1 to the group. So, our protocol requires n time slots. This leads to linear growth in the time slots required. In our Protocol, the two members can generate partial keys in each phase. One member can generate one k-exponent partial key, the other one can generate k k-exponent partial keys (total k+1 k-exponent keys) in the k^{th} phase. In CSGC-WANET-VSM framework, 3(n-4) unicasts, 1 multicast to generate partial keys for all members of the virtual subnet. So, total 3(n-4)+1 time slots required to generate group key for the virtual subnet..

In Fig.3, we plot the number of time slots required versus the number of users in the network for the proposed protocol and CSGC-WANET-VSM. As can be seen, our protocol is more efficient, this efficiency cannot depend on the group size. In Table.1, we show the number of time slots required versus the number of users in the network. By observing the table, we know that the CSGC-WANET-VSM protocol becomes impractical for large groups, our protocol scale well even the group size is very large, CSGC-WANET-VSM does not scale well with very large groups.

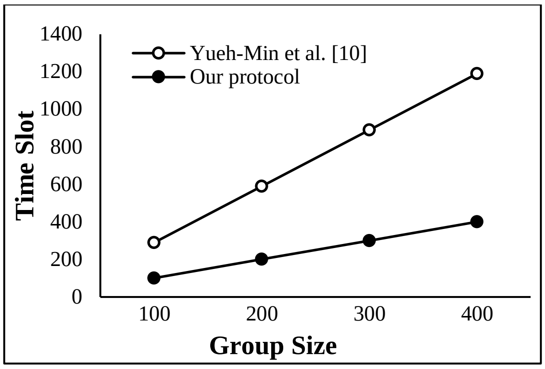

Fig.3. Comparison between CSGE-WANET-VSM and Our protocol

Table.1. Comparison between CSGE-WANET-VSM and Our protocol

Group size	CSGC-WANET-VSM	Our Protocol
500	1489	500
600	1789	600
700	2089	700
800	2389	800
900	2689	900
1000	2989	1000

6. CONCLUSION

In this article we construct a secure group communication on a virtual subnet model in a MANET, and achieve both security and efficiency for many-to-many communication. By using our framework we can create any number of subnets, an entity can be a member of one or more subnets, he needs to store separate partial key for each group (subnet) he belongs to. Now a days all networking operations (including operations on mobile devices) are performed in sharing based approach. Our framework is compatible to resource sharing in cloud computing. This is same as account sharing in online banking system, broadcast systems such as Pay-TV, Internet multicast and mobile telecommunication for a group. A node may quit or join a group over time. We developed, a sophisticated group regeneration method to facilitate this MANET feature. The main limitation of our protocol is that the dishonest entities, who are not interested to give their partial signatures to the authorised entity can be revocated from the group by using the construction and Key agreement phase.

REFERENCES

[1] V. Rajaravivarma, “Virtual Local Area Network Technology and Applications”, *Proceedings of International Symposium on System Theory*, pp. 49-52, 1997.

[2] J. Jubin and J.D. Tornow, “The DARPA Packet Radio Network Protocols”, *Proceedings of the IEEE*, Vol.75, No. 1, pp. 21-32, 1987.

[3] N.B. Salem and J.P. Hubaux, “Securing Wireless Mesh Networks”, *IEEE Transactions on Wireless Communications*, Vol. 13, No. 2, pp. 50-55, 2006.

[4] E. Shi and A. Perrig, "Designing Secure Sensor Networks", *IEEE Transactions on Wireless Communications*, Vol. 11, No. 6, pp. 38-43, 2004.

[5] A. Stojmenovic, "Geocasting with Guaranteed Delivery in Sensor Networks", *IEEE Transactions on Wireless Communications*, Vol. 11, No. 6, pp. 29-37, 2004.

[6] Y.M. Huang, H.Y. Lin and T.I. Wang, "Inter-Cluster Routing Authentication for Ad Hoc Networks by a Hierarchical Key Scheme", *Journal of Computer Science and Technology*, Vol. 21, No. 6, pp. 997-1011, 2006.

[7] IEEE Standards Association, "IEEE Standards for Local and Metropolitan Area Networks, Virtual Bridged Local Area Networks", Available at https://standards.ieee.org/standard/802_1Q-2005.html, Accessed at 2005.

[8] W. Diffie and M. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, Vol. 22, No. 6, pp. 644-654, 1976.

[9] M. Steiner, G. Tsudik and M. Waidner. "Diffie-Hellman Key Distribution Extended to Group Communication", *Proceedings of ACM Conference on Computer and Communications Security*, pp. 31-37, 1996.

[10] Yueh-Min Huang, Ching-Hung Yeh, and Tzone-I Wang, "Constructing Secure Group Communication over Wireless Ad-Hoc Networks based on a Virtual Subnet", *IEEE Transactions on Wireless Communications*, Vol. 14, no. 5, pp. 70-75, 2007.

[11] T. Karthikeyan and K. Praghash, "An Improved Task Allocation Scheme in Serverless Computing Using Gray Wolf Optimization (GWO) Based Reinforcement Learning (RIL) Approach", *Wireless Personal Communications*, Vol. 117, No. 3, pp. 1-19, 2020.

[12] S. Kannan, G. Dhiman, and M. Gheisari, "Ubiquitous Vehicular Ad-Hoc Network Computing using Deep Neural Network with IoT-Based Bat Agents for Traffic Management", *Electronics*, Vol. 10, no. 7, pp. 785-796, 2021.

[13] N.G. Veerappan Kousik, K. Suresh, R. Patan and A.H. Gandomi, "Improving Power and Resource Management in Heterogeneous Downlink OFDMA Networks", *Information*, Vol. 11, No. 4, pp. 203-216, 2020.