Cluster Based Composite Key Management in Mobile Ad Hoc Networks

R.PushpaLakshmi
Assistant Professor, IT
PSNA College of Engineering & Techchnology, Dindigul
Tamilnadu, India.

Dr.A.Vincent Antony Kumar
Professor & Head, IT
PSNA College of Engineering & Techchnology, Dindigul
Tamilnadu, India.

ABSTRACT
Mobile ad hoc network is a wireless network without the support of network infrastructure. Security is one of the main challenge in ad hoc network due to dynamic topology and mobility of nodes. In this paper, we present a new composite key management scheme based on a combination of techniques such as hierarchical clustering, partially distributed key management, offline certification authority and mobile agent. In this method, network is hierarchically structured into clusters based on node's trust value. The information about node revocation and PKG service nodes are exchanged in the network using mobile agents. The cluster head maintains public keys of its members which overcomes the storage problem in PKI where each node maintains public key of other nodes in network and avoids centralized CA to generate keys, thus enhances security.

Categories and Subject Descriptors
[**Network Security**]: Key management schemes – *composite Key management, certification authority, trust evaluation*.

General Terms
Network security.

Keywords
Ad hoc network, trust value, mobile agent, key management, hierarchical clustering, certificate authority, network security.

1. INTRODUCTION
Ad hoc network consists of mobile nodes which communicate with each other through wireless medium without any fixed infrastructure. MANET consists of wireless nodes that move dynamically without any boundary limitation. Mobile ad hoc network are more prone to security threats and attacks. Key management is the main problem in ad hoc network as the nodes move freely in the network and have limited memory. Symmetric key systems uses the common key for both encryption and decryption. This method is faster, easier to implement, and it lowers overhead on system resources. A major disadvantage of symmetric algorithm is the exchange of shared secret key between two parties. This security mechanism lacks in data authentication and integrity.

Public key cryptography uses two keys: public key and private key to establish secure communication, in which the sender and receiver doesn't require to share a secret key. This security mechanism ensures data authentication and confidentiality without shared secrets.

To ensure secure communication, the selected cryptography algorithm must have following requirements:

confidentiality – Protecting the data from all nodes except the intended receiver.

Authentication – Proving one's identity.

Integrity – Ensuring no unauthorized alteration of data.

Non repudiation – Preventing an entity from denying previous commitments or actions.

2. RELATED WORK
2.1 Partially Distributed Certificate Authority
The scheme is suitable for planned, long term ad hoc network. This scheme is based on public key encryption. The method uses trusted offline CA and (k,n) threshold scheme to protect private key[1]. Offline dealer assigns a valid certificate and public key to the node that join the network. The private key of the node is shared by k serving nodes. The serving nodes are selected randomly in the network. The new node must collect all the n partial key shares to compute the whole private key. This scheme has the following drawbacks: i) serving node must maintain public key of all other nodes in the network, which requires more memory space ii) lack of certificate revocation mechanism iii) not suitable for larger network iv) the algorithm doesn't deal with network synchronization when split or join occurs in the network v) serving nodes may not be in contact at all times[10].

2.2 Fully Distributed Certificate Authority
The scheme is based on public key encryption and is suitable for long term ad hoc network. Unlike partially distributed CA, the capability of certificate authority is distributed to all the nodes in the network[7][8]. All nodes in network holds partial share of the private key. Private key is computed by combining any k partial shares. This scheme has the following drawbacks: i) the method doesn't deal with network synchronization ii) threshold parameter k need to be larger since attacker may compromise large number of shares between share update iii) complex maintenance protocol.

2.3 Identity based key management scheme
The scheme uses set of Private Key Generation(PKG) nodes to generate public key and private key of the node. The public key of the node is generated based on node's identity. A node must contact at least k PKG nodes to obtain its private key. This scheme reduces communication and computation cost because each node would not have to create its own public key and

broadcast it in the network. This scheme doesn't deal with key update[4][10].

2.4 Self Issued Certificates

The scheme is suitable for long term network that does not require any infrastructure. There is no centralized CA. Each node create its own private key and certify public key to other nodes, if it has trust on that node[2][3]. This scheme has the following drawbacks: i) the method doesn't deal with certificate revocation ii) during initial stage, the certificate chain may not be found between all nodes in the network. iii) the system is less trusted without any trusted authority[10].

2.5 Secure Pebble nets

Secure pebble net is suitable for long term ad hoc network with low performance nodes. Network is partitioned into pebbles, where node with maximum weight is selected as key manager. All nodes in a pebble share a common traffic encryption key for secure communication[5]. This scheme support group authentication and not support individual member authentication.

3. COMPOSITE KEY MANAGEMENT SCHEME

3.1 Network Model

The network is partitioned into clusters. The node with maximum trust ability is selected as CH. Among cluster members in a cluster, k nodes with high trust value are selected as PKG serving nodes. Nodes can move from one cluster to another. Network administrator selects CH. The offline CA assigns node_id for the node that join the network. Each new node has a self assigned public key and register its information in CH. The private key of the node is generated by PKG serving nodes. The CH also acts as one of the PKG serving nodes and play the role of key combiner. Private key shares generated by k PKG serving nodes are combined by CH to obtain whole key. Initial public key of CH is obtained by applying one way hash function on its id. CH public key varies based on its trust level. New public key of CH is computed based on old pubic key and its new trust value. The private key of CH is initially assigned by network administrator. Later private key shares are computed by PKG nodes. The public key of CH is distributed to all cluster members in the corresponding cluster. The mobile agent is a program segment that collects information about k trustable nodes in a cluster and information about nodes whose certificate is revoked[6]. The system use two level of frequency for communication: low level frequency is used to establish communication between cluster members and high level frequency is used to establish communication between cluster heads.

3.2 Hierarchical clustering algorithm

We apply the concept of dominating set based clustering for partitioning the network into clusters. The dominator node is selected based on two factors: trust ability of the node, probability of future contact of the node. The node with maximum trust ability is elected as CH. All other nodes lies within the transmission range of CH acts as cluster member.

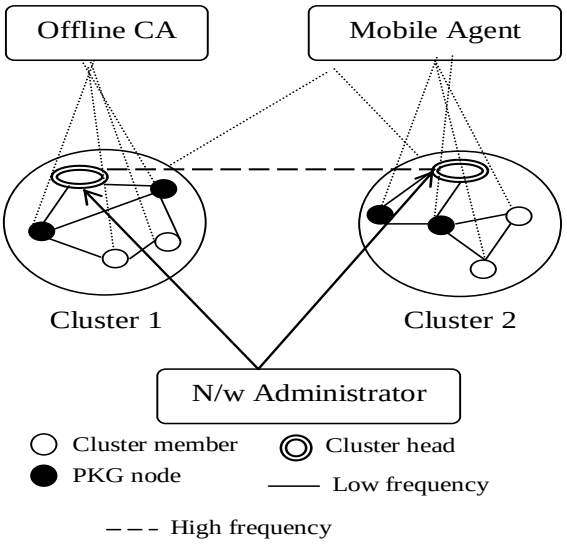

Figure 1. Network Model

3.2.1 Dominator Election Algorithm

Dominator election algorithm selects a node having maximum trust ability and maximum probability of future contact as dominator. The basic algorithm is similar to Wu and Li algorithm. The algorithm computes trust value of each node based on node's neighbors opinion. For node i in the network the algorithm compares node's trust value with trust value of its neighbors. The resultant node with maximum trust ability is marked as dominator. If there exists nodes with equal trust ability, the algorithm compares the probability of future contact of nodes with neighbors. It select a node with maximum probability as dominator. A node is selected as dominator only if it is not in dominating set and N(i) not in dominating set. The selection mechanism is performed when any of the following happens: CH goes down due to low battery, CH moves outside the cluster, periodically for certain time interval. The procedure for dominator election is described as follows.

```
For each node i in the network
1.      Compute Ti  = compute_trust(i, N(i))
End for
For each node i in the network
        For each node j in N(i)
2.        Find node k with max(Ti,Tj)
3.        If  Ti = Tj
            3.a.  Calculate Pcontact for i and j
            3.b.  Find node k with max(Pcontact )
          End if
4.      Add k  to DS if k ϵ DS and N(k) C DS
        End for
End for
```

3.2.2 Trust value Evaluation

When a node send a packet, it updates the packet forwarding status of its neighbors in status table. Forwarded and unaltered fields in status table are set as 1 if the packet is forwarded successfully without any error, else set as 0. Apply AND operation on forwarded and unaltered fields of each neighbor[9]. Count the number of 1 on the resultant value. The probability of successful transmission is the ratio between count and number of forwarded packets. If it is greater than 50% increment trust value of neighbor by 1 else decrement trust value by 1. Set trustworthy as 'Yes' if trust value >= threshold , else set as 'No'. The trust value of a node is calculated by combining the node's neighbors opinion about its trust ability. The procedure for evaluating trust value of a node is described as follows.

```
Compute_trust(node i, N(i))
{
1.      Trustcnt=0
2.      Non=0
3.      For each node j in N(i)
        3.a.    If trustworthy = 'Yes'
                3.a.a   Trustcnt = Trustcnt+1
                3.a.b.  Non = Non +1
                End if
        End for
4.      Return(trustcnt/non);
}
```

3.2.3 Probability of future contact evaluation

Each node maintains details about the contact with its neighbors in contact table. Probability of future contact of a node is computed based on duration of previous contact and total number of previous contact. The procedure to compute probability of future contact of a node is described as follows.

```
Prob_contact(node i, N(i))
Begin
        For each node j in N(i)
                                    tij
        Pcontact(i) = ∑  ------------------  * NCij
                                ( ti + tj ) /2

        End for
end
```

3.3 Key Generation

The public key of CH changes with respect to its trust value. The public key is evaluated based on its previous public key and its trust value. Each new node maintains public key which is self assigned. To obtain private key, new node request PKG nodes for its key shares. Nodes with highest trust value next to CH is selected as PKG serving nodes. The information about PKG nodes is exchanged across the network through the mobile agent. CH broadcast its public key to all nodes in the cluster.

$$P_{CHi} = \begin{cases} H(ID), \text{ initial public key} \\ f(\text{Prev. } P_{CHi} \| \text{ trust value}) \end{cases}$$

P_n = self assigned string

$S_n = S_{PKG1} + S_{PKG2} + \ldots + S_{PKGk}$

3.4 Node Join

Offline dealer assigns unique id for the new node. The new node register its public key information in CH. CH records the information about new member in its member table with fields: mem_id, public key. New node can contact CH either directly or through intermediate cluster members.

1. $N \rightarrow CH_i : E_{PCHi} (REQ(ID_N \| R))$
2. $CH_i \rightarrow PKG : E_{PKG1} (REQ(ID_N \| R'))$

   $\ldots\ldots E_{PKGk} (REQ(ID_N \| R'))$
3. $PKG \rightarrow CH_i : E_{PCHi} (S_{P1} \| R')) \ldots E_{PCHi} (S_{PK} \| R'))$
4. CH computes private key

   $S_N = S_{PKG1} + S_{PKG2} + \ldots\ldots + S_{PKGk}$
5. $CH_i \rightarrow N : E_{PN} (S_N \| R \| KNO))$ where

   KNO=Unique no. || time of expiry

Step 1: Node N send a request message encrypted using public key of CH. The message includes node id and a random number R. The random provides authentication and avoids replay attack.

Step 2: CH send the request message to the PKG nodes PKG_1, PKG_2,…, PKG_K. The message is encrypted using PKG's public key. R′ represents the random number used by PKG nodes for authentication.

Step 3: PKG nodes send the CH's public key encrypted partial key share to CH.

Step 4: CH acts as key combiner, generates the whole key S_N.

Step 5: CH send the generated secret key to node , encrypted using node's public key. KNO is generated using unique number and expiry time of the key.

3.5 Intra Cluster Communication

Nodes within the same cluster communicate using low frequency range. Assume node i wants to communicate with node j of same cluster. The algorithm for intra cluster communication is as follows.

1. i $\longrightarrow$ CH : E_{PCH} ($ID_i \parallel ID_j \parallel R$)
2. CH $\longrightarrow$ i : E_{Pi} ($P_j \parallel R$)
3. i $\longrightarrow$ j : E_{Pj} ($ID_i \parallel R' \parallel g^i$)
4. j $\longrightarrow$ i : E_{Pi} ($ID_j \parallel R' \parallel g^j$)
5. Node i and j generate a session key SK using g^i and g^j.
6. i $\longrightarrow$ j : E_{SK} (M)

Step 1: Node i request CH for node j's public key. The request is encrypted using CH's public key.

Step 2: CH send the private key of node j encrypted by node i public key, to node i.

Step 3: Node i introduce itself to node j by sending message encrypted using j's public key. g^i is random generator assigned by node i.

Step 4: Node j send a reply message encrypted using public key of node i. g^j is random generator assigned by node j.

Step 5. Using g^i and g^j , node i and j generate a session key for secure communication.

Step 6: Node i exchange message with node j which is encrypted using session key.

3.6 Inter Cluster Communication

Nodes in different clusters communicate through CH. CHs communicate with each other using high frequency. Assume node i is in C1 and node j is in C2. Node i wants to communicate with node j. The algorithm for inter cluster communication is as follows.

1. i $\longrightarrow$ CH1 : E_{PCH1} ($ID_i \parallel ID_j \parallel R$)
2. CH1 $\longrightarrow$ Backbone : REQ($ID_i \parallel$ CH1)
3. CH2 $\longrightarrow$ CH1 : E_{PCH1} (CH2 $\parallel R' \parallel g^a$)
4. CH1 $\longrightarrow$ CH2 : E_{PCH2} (CH1 $\parallel R' \parallel g^b$)
5. CH2 $\longrightarrow$ CH1 : E_{SK} (P_j)
6. CH1 $\longrightarrow$ i : E_{Pi} ($P_j \parallel R$)
7. i $\longrightarrow$ CH1 : E_{PCH1} (E_{Pj} (M))
8. CH1 $\longrightarrow$ CH2 : E_{PCH2} (E_{Pj} (M))
9. CH2 $\longrightarrow$ j : E_{Pj} (E_{Pj} (M))

Step 1: Node i request CH1 for node j's public key. The request is encrypted using CH's public key.

Step 2: CH1 forwards the request message to all other CHs.

Step 3: If node j is member of CH2, CH2 send it identification to CH1, encrypted by CH2's public key. R′ represents a random number for authentication. g^a is random generator assigned by CH1.

Step 4: CH1 respond CH2 with its identification and g^b is random generator assigned by CH2.

Step 5. Using g^a and g^b , node CH1 and CH2 generate a session key for secure communication. CH2 send CH1 the public key of node j, encrypted by session key.

Step 6: CH1 forwards the node j's public key to node i, encrypted by public key of node i.

Step 7: Node i send the message encrypted by j's public key, which in turn encrypted by CH1's public key.

Step 8: CH1 forwards the encrypted message to CH2, encrypted using CH2's public key.

3.7 Node Leave

When a node leave a cluster, the CH removes the node information from it's member list. The key of CH changes continuously with respect to its trust value. So the node that left the cluster cannot recover the new messages using the old public key of cluster head. Similarly when the node that joins the cluster again cannot recover previous messages, because the new key differs from old key. Thus the system maintains forward secrecy and backward secrecy.

3.8 CH leave

The CH before it leaves the cluster, send a LEAVE message to PKG nodes. The PKG node with next highest trust ability plays the role of new CH. The old CH send information about all its cluster members to new CH. The new CH broadcast (oldCH_ID, newCH_ID, new public key) to all other CHs. The CH periodically send a REFRESH message to all PKG nodes. If any PKG node doesn't receive the REFRESH message within specified time, it contact other PKG nodes and declare that the CH had left the cluster. New CH is elected based on trust value.

3.9 Key Update and Renewal

The node can request CH for a key renewal before its certification time expires. If the certification time not expired, the CH extend the time period of the key. If time expired, the node must request for key update to PKG nodes.

1. N $\longrightarrow$ CH : E_{PCH} (KEY_RENEWAL(R $\parallel$ KNO))
2. CH $\longrightarrow$ N : E_{PN} (KNO$\parallel$R) // if time not expired.
3. If time expired, follow the procedure for new key generation.

Step 1: Node send a KEY_RENEWAL message to CH, encrypted by CH's public key. R represents random number for authentication. KNO represents key number.

Step 2: If certification time is not expired, the CH respond the node with the renewed certificate, encrypted by node's public key.

Step 3: If the node's certification is expired, then the node must request for new key.

3.10 Key Revocation

Mobile agent is a program segment which run on all nodes in cluster. Mobile agent maintains a data structure which includes two fields: node ID, revoke point. Initially, revoke point is set as 0 for all nodes in cluster. The node that suspect a particular node, can update the entry for that node in data structure by incrementing its revoke point value by 1. The information provided by mobile agent data structure will be processed by CH. The certificate of the node with revoke point greater than or equal to threshold will be revoked by CH. Finally the CH broadcasts REVOKE(ID) message to all other nodes in the cluster.

Table 1. Information in data structure.

Node _ID	Revoke Point
1	0
2	0

3.11 Network Split and Join

When network partition occurs, consider the following cases:

- New partition not includes any CH and PKG nodes- Elect new CH and PKG nodes for the partition.
- CH remains in the same sub network – Select new PKG nodes in the sub network.
- PKG nodes remains in the same sub network – Select the PKG node with highest trust value as CH.
- Both CH and PKG remains in the sub network – Update the member list with the new member details.

When network join occurs, compare the CH trust values and select a CH with the highest trust value as a new CH. Similarly, PKG nodes are selected by comparing its trust value.

4. EXPERIMENT RESULTS

In this paper, we presented a secure composite key management for mobile ad hoc network, which improves network performance and reduce message overhead. In this section, we present simulation results for secure routing through trustable nodes based on AODV protocol. In our future work, we will evaluate the performance of composite key management by simulation and compare it with that of other existing key management schemes.

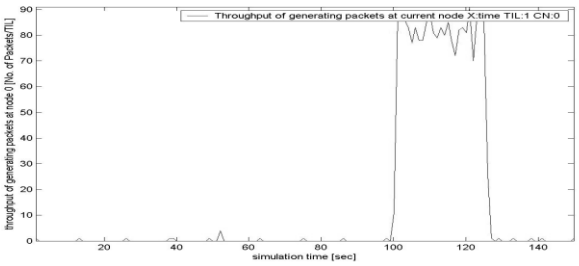

Figure 2. Throughput of generating packets relative to simulation time.

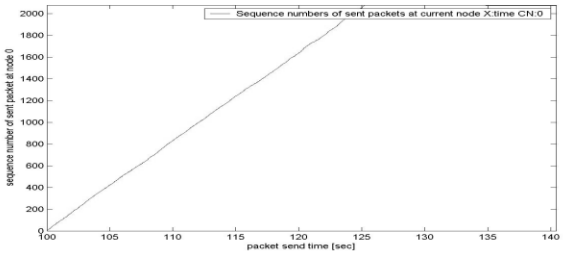

Figure 3. Sequence no. of sent packets relative to packet send time.

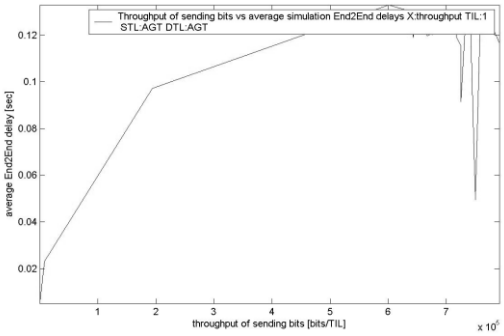

Figure 4. Throughput of sending bits relative to average end to end delay.

5. CONCLUSION

In this paper, we presented a composite key management scheme for mobile ad hoc network. The public key of every cluster head is computed based on the old public key and the current trust value. Each cluster node has its self assigned public key, which is known only to the cluster head. Cluster nodes don't need to broadcast their public key to all the cluster nodes, which saves network bandwidth and storage space. The private key of the cluster node is computed using the partial keys generated by the k PKG nodes. Each key has unique key number which includes key id and timestamp. Key renewal process can be done easily using the timestamp in key number. Mobile agent, which runs periodically in cluster nodes, handles key revocation process and PKG nodes selection process. The scheme uses hierarchical clustering method, which supports network extendibility.

6. REFERENCES

[1] Zhou, L. and Haas, Z.J. 1999. Securing Ad Hoc Network. IEEE Networks, Vol. 13, No. 6.

[2] Hubaux, J.P., Buttyan, L. and Capkun, S. 2003. Self-Organized Public-Key Management for Mobile Ad Hoc Networks. Transactions on Mobile Computing.

[3] J-P. Hubaux, L. Buttyán and S. Capkun. 2001.The Quest for Security in Mobile Ad Hoc Net- works, ACM.

[4] Zheng, S.R. 2004. The Technique of Ad Hoc Network, Posts & Telecom press.

[5] Stefano Basagni, Kris Herrin, Danilo Bruschi, Emilia Rosti. 2001. Secure Pebblenets, ACM.

[6] ZHANG Yi, ZHU Lina and FENG Li.2009. Key Management and Authentication in Ad Hoc Network based on Mobile Agent JOURNAL OF NETWORKS, VOL. 4, NO. 6, AUGUST.

[7] Seyed-Mohsen Ghoreishi, Morteza Analoui. 2009. Design a secure composite key-management scheme in Ad-Hoc Networks using Localization IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.9, September.

[8] H. Luo, P. Zerfos, J. Kong, S. Lu and L. Zhang. 2002. Self-securing Ad Hoc Wireless Net- works, IEEE ISCC.

[9] Huaizhi Li, Mukesh Singhla. 2006. A secure Routing protocol for Wireless Ad hoc Networks, Proceedings of the 39th Hawwaii International Conference on System Science.

[10] Yingfang Fu1, Jingsha He2, and Guorui Lil. 2006. A Composite Key Management Scheme for Mobile Ad Hoc Networks, OTM Workshops 2006, LNCS 4277, pp. 575 – 584.