International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 4– No.6, November 2012 – www.ijais.org

Distributed Certificate Management in Mobile Ad Hoc Networks

Mohammad Masdari
Computer Engineering Department, Islamic Azad University, Urmia Branch, Urmia, Iran,

Javad Pashaei Barbin
Computer Engineering Department, Islamic Azad University, Naghadeh Branch, Naghadeh, Iran,

ABSTRACT
PKI or public key infrastructure is used many security solutions that are designed for mobile ad hoc networks. These networks have special features that distinguish them from other wired and conventional networks and centralized Certificate Authorities cannot be used for certificate management in these kinds of networks. Thus many efforts have been made to adapt Certificate Authority's (CA) tasks to the dynamic environments of MANETs and distribute the tasks of CA among MANET nodes. In this paper, we study various Certificate management solutions that are proposed in the literature and analyze their advantages and limitations. In addition, we emphasis on certificate revocation and validation issues and compare the overheads of these operations. Finally, we propose the characteristics of an ideal DCA system that can be used to verify the completeness of any DCA Scheme.

Keywords
Security, Distributed Certificate Authority, Threshold cryptography, Digital Signature.

1. INTRODUCTION
Mobile ad hoc networks or MANETs are vulnerable to various passive and active security attacks that are launched by internal and external attackers. But because of special characteristics of MANETs, such as lack of any fixed infrastructure, mobility of nodes and limited bandwidth of wireless communication, establishing security in MANETs is a challenging issue. Numerous solutions have been designed and presented in the literature to increase the security level of these networks. Public key cryptography is used to provide privacy, integrity, authentication and other security services in Internet and other conventional networks. Certificates are on the main security data structures of PKI systems that assure the authenticity and integrity of public keys. Certificate authorities are trusted third parties that are used for issuing, revoking and managing of user certificates. But, MANETs lacks any infrastructure and are created dynamically by cooperation of mobile and wireless devices. Thus, for adapting PKI to mobile ad hoc networks, the tasks of certificate authority (CA) should be distributed on the user nodes or the functionality of CA should be emulated somehow. To solve all of these problems, the following certificate management solutions have been proposed in the literature:

- Web of trust-based schemes.
- Certificate Authority-based schemes.

In the first case, every entity certifies the binding of identities and public-keys for other entities. In the second case, certificate are issued and managed by a certificate authority. Generally, the following two kinds of CA are used in the ad hoc networks:

- Dependent CAs
- Independent CAs

Dependent CAs are used in the hybrid MANETs and certificate authority may depend on the centralized CA that resides on a fixed network. These, CAs act as a front-end to the main CA and support the MANET users' requests by contacting the main CA. The independent CAs are useful in non-hybrid MANETs which are not connected to the fixed and wired networks. These distributed CAs are created dynamically and perform their operations distributedly by cooperation of ad hoc network nodes. In a Distributed Certificate Authority (DCA) private key is distributed among the shareholder nodes. When operations such as issuing or revoking certificates are required, threshold of shareholders participate to perform the requested service [4]. Like conventional CA, the public key of the DCA will be known by all network's nodes and will be used to verify the signatures of certificates issued by the DCA. Many distributed certificate authorities schemes have been designed for MANET and they can be classified as the following items:

- Partially distributed certificate authority
- Fully distributed certificate authority

In PDCAs the services of CA are distributed to a set of specialized server nodes using secret sharing. Each of these nodes can generate partial certificates and a client can create a valid certificate by combining enough number of these partial certificates. Therefore, PDCAs specially are useful in heterogonous MANETs which consist of some special nodes that have more processing and communication capabilities. However, in homogenous MANETs which all nodes are identical, the nodes of the distributed CA might be chosen randomly. In fully distributed certificate authorities or FDCA, all nodes became the DCA share holder and can generate partial certificates [15]. FDCA reduces the communication delay and improves the availability because almost all neighbors of a requesting node hold shares of the DCA's private signature key. However, it is more vulnerable to malicious nodes and behaviors,
because more nodes are part of the DCA. To overcome this problem, some schemes use trust management systems in combination with DCA systems and some other schemes use intrusion detection system to monitor the ad hoc network. Almost all FDCA and PDCA schemes use threshold cryptography which requires cooperation of k nodes from total n nodes. Thus k contact is need for each operation in the DCA which means that a client needs to contact at least k CA nodes and receive at least k replies. However it is possible that more than k, CA node receive the certificate request and respond to it, thus a client receives more responses than it needs. In [16], Luo et al, add the following items as properties of a distributed CA:

- **Liveness**: The CA always processes a request in a finite amount of time.
- **Safety**: An adversary is never able to forge a certificate.

- **Freshness**: A query to the CA returns the most recent status of a targeted certificate.

Table 1 compares the features of various certificate management methods in MANETs. In this paper we analyze various certificate management schemes in the mobile ad hoc networks and study the techniques and solutions which have been used in them. The rest of the paper is organized as follows: In section 2, architecture of distributed certificate authorities in MANET are specified. In section 3, the tasks of DCA system is discussed and the certificate request protocol for DCA schemes are illustrated section 4. At last, in section 5, we present the properties of an ideal DCA system for MANET.

2. DCA Architecture

Architecture of a DCA system specifies its consisting components that are needed for proper operation of a DCA system. It also determines the functions and tasks of MANET nodes in a distributed certificate authority. Different schemes use different kinds of nodes for providing PKI services, but generally we may have the following nodes in DCA schemes:

- DCA share holders
- Repositories
- User nodes

DCA schemes that are designed for large MANETs use one or more repository nodes. The repository nodes store certificates, CRLs and other PKI related data structures. Often DCA schemes use special nodes with higher capabilities as DCA server nodes or clusterheads for this purpose. However, there are schemes that use ordinary nodes for performing this task. DCA schemes that support low number of user nodes do not use special nodes for repository and DCA share holder or user nodes store their certificates themselves. Finally, although using multiple repositories increases the availability and fault tolerance of PKI system, it makes the repositories more vulnerable to security attacks. Thus their security should be established by monitoring and detecting compromised repository nodes. More complex DCA systems use more sophisticated structure, fox example in [8] DCA consists of three tiers. At the lowest tier individual nodes are organized into clusters. The next tier consists of one or more certificate repositories in each cluster. The top tier consists of DCA servers. Within each cluster, a fixed number t of nodes are designated as repositories that store the certificates of the nodes within the cluster, the certificates of all servers, the counter-certificates of the network nodes, and the most recent version of the CRL. The repositories might also become compromised and thus become unavailable. However, up to r repositories may be compromised within a cluster before a new node within a cluster is elected to serve as a repository. This way, there will always be a minimum of t - r active repositories in each cluster. In [6], the system architecture contains three types of nodes: an administrator which will be present only at the initialization step, it should play the role of a certification authority for the existing clusterheads, to certify existing clusterheads, distribute his secret key over them according to the secret sharing scheme and finally give them his certificate. Each node has a private and public key. In this architecture, we consider that each clusterhead is a central certification authority for its cluster members and that it is initialized either by the administrator or by a coalition of K other clusterheads. The administrator responsibilities are distributed over the existing clusterheads. Therefore, every operation requiring the administrator private key SKCA (Central Authority Secret Key) can be accomplished by any K collaborating clusterheads. Thus, every clusterhead will be supplied with a partial key of the certification authority secret key, a valid certificate and the administrator certificate. Clusterheads will be then considered as a distributed certification authority for the new ones. In [2], network nodes structure into clusters. At the inception, n nodes have been selected as CA server nodes and each of them is assigned with one share by applying the (t,n) secret sharing scheme. If a CA node be at the cluster, it will be clusterhead in that cluster. Otherwise the CH is only delegated to managing and distributing CA information, but will not participate in CA share updates. In [16], DCA system is consisted of three types of nodes: clients, servers and the mCA which is connected to the backbone. mCA acts as the offline CA and assigns a set of N special nodes to constitute a distributed CA or dCA that performs the role of the online RA. The mCA controls the admission of a client/server node to the network at its command, through the issuance of a certificate that asserts the binding between the identity and initial public key of the node. When the network is disconnected from the mCA, clients submit their requests to the dCA. Also, in some schemes such as [7] once the cluster is formed, the cluster head acts as a certification authority for all its members and cluster members act as user nodes.

3.Tasks of a DCA system

In this section, we specify the operations that are needed to initialize and maintain a DCA system in mobile ad hoc networks. For providing PKI service in MANET, each DCA scheme should supports two kinds of operations:

- Operations that are needed for maintaining the DCA and are performed by cooperation of DCA members or Intra-DCA operations.
- Operations for supporting user nodes.
  For supporting DCA member nodes each scheme require the following operations:
- Private and public key generation for DCA share holders.
- Joining and leaving to DCA system
- Secret sharing
- Private key refresh
- Certificate Creation for DCA
- Tuning the threshold value
- Handling network partitioning
- Support of a consistent certificate revocation scheme
- Secure intra-DCA communication
- Locating other DCA nodes
- Finding compromised DCA members
- Cooperation in producing the CRLs

For supporting MANET users and providing certificate related services, each DCA system should support the following operations:

Table 1: Comparison of Certificate Management in Mobile Ad Hoc Networks			
	Centralized CA	**Distributed CA**	**Web of trust**
Security	High	Low	High
Availability	Low	High	Low
Fault tolerance	Low	High	Low
Messaging overhead	Low	High	
Performance	High	Low	Low
Message exchange	Low	High	High
Scalability	High	Low	Low
Routing dependent	No	Some schemes	No
special nodes	required	Only PDCA	Not required
Revocation source	Owner, Issuer	Owner, issuer, k accusation	Issuer
Validity of certificate	High	Low	

- Joining new users and creating private and public keys.
- Requesting certificate by a certificate request protocol.
- Certificate renewal.
- Revocation of malicious user node's certificate
- User node mobility.
- Authentication of user nodes.

For providing a scalable and efficient DCA system, all of these operations should be implemented by low overhead solutions.

Because a MANET is a dynamic environment, a DCA scheme must support joining and leaving of nodes to DCA system. Also, because all certificate authority's tasks are performed distributedly by multiple nodes, each operation must be done with cooperation of subset of MANET nodes. Thus each scheme should present efficient and consistent solutions for performing each task by a distributed algorithm.

4.Certificate Request protocol

User nodes should send their requests to DCA scheme to create a certificate distributedly. Each scheme provides a certificate request protocol that is adapted to its architecture and properties. Generally, a reliable and efficient certificate request protocol should consider the following issues:

- How many request messages should be sent for each requested certificate in a threshold cryptography scheme?
- To which DCA servers, request should be sent, if more than K DCA member is founded?
- How these requests must be distributed, using multicast, unicast or multipath protocols?
- How much we should wait for response of requests? Which parameters should be used for calculating timeout value? Congestion, link failures, mobility?
- What request transmission scheduling should be used for prevention of request implosion?
- When multiple nodes want to send their requests, what retransmission scheduling should be used?
- Where the certificate of users and DCA system will be stored?
- How far the requests will be distributed? For example requests can be limited by TTL or by GPS and localization methods.
- How we should locate the DCA nodes?
- Who should locate the DCA nodes?
- How DCA location information should be found? Push based or pull based? Or by autoconfiguration methods?

The question is that how many request messages should be send for each certificate? The simplest method is flooding. But it generates large traffics and since a client has no way to limit the CREQ, all the MOCAs receive a CREQ and respond with a CREP. Any partial signatures beyond the required *k* are discarded and waste networking and processing resources [10]. In addition, if few CREPs are received, the client's CREQ timer expires and the certification request fails. On failure, the client can retry or proceed without the certification service. Thus some schemes propose solutions to solve this problem. In [10], client nodes use the MP or MOCA certification Protocol for contacting sufficient MOCAs. In MP, the client waits a fixed period of time for *k* such CREPs. MP use β-unicast, where the client applies multiple unicast connections to replace flooding. The node should send out additional CREQs to increase the probability of success. The number of additional CREQs is defined by α, a marginal safety value used to increase the success ratio of β-unicast. *It* can be determined based on the node's perception of the network status. Generally, a client should send request to K DCA member, but what should we do if we found more than K DCA member is found? In [10], if there are more than β routes in the cache, the choice of which ones to use can affect performance. We define three different schemes:

1. Random MOCAs - Choose β random MOCAs with cached routes.
2. Closest MOCAs - Choose β MOCAs with smallest hop counts in the cache. This benefits of the shortest response time and the smallest packet overheads.
3. Freshest MOCAs - Choose β MOCAs with the freshest cache entries. The most recently added or updated entries should not be stale, especially under high mobility. By choosing the freshest MOCAs, the client should be able to minimize the risk of failure under high mobility.

In [11], every such CREQ is associated with a timer. Any CA node that receives such a CREQ message, replies with a *Certificate Reply* (CREP) message that contains the partial signature of the CA node. If the client node is successful in receiving *k* valid CREPs, it constructs the signature otherwise, the certification process fails and client has to start the process again. The CREQ and CREP messages can be piggybacked on the routing packets thereby reducing the communication overhead of the protocol. For propagation of the CREQ messages flooding or multiple CREQ message can be used.

4.1. Locating CAs

One of the important phases of each certificate request protocol is the efficient locating of DCA nodes. We need DCA nodes locating in the following issues:

- When a DCA member want to cooperate with other DCA nodes for providing PKI services to user nodes.
- When a user node want to send its requests to multiple DCA members.

Generally, in a DCA locating algorithm we have to determine the following items:

- Who should locate the DCA nodes?
- How DCA location information should be found? By push based or pull based? By autoconfiguration system or name systems?

- How this information must be passed to the other nodes?

In [2], each clusterhead maintains CA information table, which contains a list of the CA nodes in its local cluster, and probably the CA information in other clusters. For locating DCA nodes it sends a request to cluster head to get CA Information. Then, cluster head collects CA distribution information, and passes it to the user. In [17], to maintain the registration table of bindings of IP addresses and public keys, the server nodes need to track of each other with the periodic HELLO messages. Otherwise, if all the server nodes leave without notice, the registration information will be lost, which will lead to high communication overhead in the subsequent registrations. Name systems are one of the components that can be used in locating the DCA members. In [21] we have studied various name systems that are proposed in the literature. Name systems can be used for locating DCA members but no scheme has proposed to use them. Also, in schemes that network nodes are equipped with GPS devices, geographical routing protocols can be used for limiting the request distributions.

4.2.Load Balancing

In large MANETs, network consists of many nodes which will contact the DCA members for various operations such as requesting certificate or other issues. It is ideal that load balancing be considered in communication such as certificate request protocols. In this case some method should be used to prevent saturation of DCA members under heavy loads. For example user nodes can use different DCA members in each request. However, this solution is applicable in dense MANETs that user nodes have numerous connections to its neighboring nodes. For more intelligent solutions, DCA members should inform their status in some advertisement messages. But these messages will put some overheads on ad hoc networks. One of the techniques that can be used to recognize the load of DCA members is to monitor the traffic of DCA members by their neighboring nodes. This technique has not messaging overheads but it increases the processing overheads of some nodes. These nodes can redirect certificate requests to other DCA members if they found the neighboring DCA members are busy.

5.Certificate Revocation

As figure 1 shows numerous schemes that have been presented for certificate revocation and validation purpose in literature. But these scheme, does not state that how the revocation and validation information are gathered. A DCA scheme must provide an efficient and low overhead solution for consistent revocation of user certificates. This requires intra DCA communication and cooperation of user nodes and DCA members.

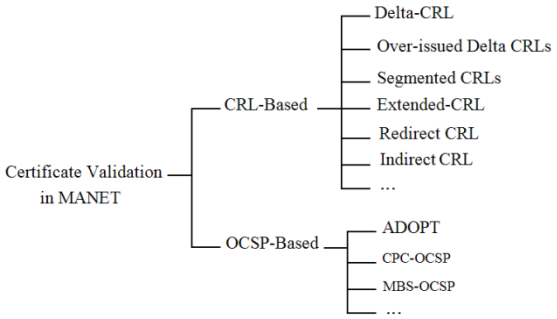

Figure 1: Certificate validation and revocation schemes in MANETs

We consider following three phases for certificate revocation process:

- Pre-revocation phase
- Revocation phase
- Post-revocation phase

Pre-revocation phase is occurred before revocation of a certificate and the following questions should be answered for this phase:

- How to detect the malicious nodes?
- How to deal with false accusations?
- Who should detect malicious nodes?
- What to do next?

A DCA scheme may propose a new solution for this problem or it may use other existing methods. Generally for detecting malicious nodes, the following methods may be used:

- Network monitoring solutions.
- Misbehavior reports of user nodes.
- Trust management systems.

The second method or misbehavior reports are vulnerable to lying attacks and some of the researchers have shown that if the number of attackers be larger than some threshold, it cannot be prevented. In such situations, a valid user can be accused by attacker nodes and its certificate may be revoked mistakenly. The third solution is the combination of two before mentioned methods and it is either susceptible to lying attacks. Numerous solution, have been presented to decrease the impact of attacks on report-based methods. For example, reports can be digitally signed or only accepted from only special nodes which have high reputation in ad hoc network. After malicious nodes have been found, a DCA system must present solutions for the following items at the revocation phase:

- How a DCA system should revoke a certificate? These include disseminating the revocation information in DCA system.
- Who should initiate the revocation process?

After revocation of malicious node's certificate, a DCA scheme must present some solution for the following question at post-revocation phase:

- How to inform users of revoked certificates, and which method should be used? Push based, Pull based or Hybrid methods?
- How user nodes can verify some certificate?

Thus a user node should have a mechanism for verification of certificate of other nodes. For the certificate verification, OCSP or CRL based methods can used but DCA system should specify that how it support these verification techniques. Most common method is certificate revocation list (CRL) that consists of a list of revoked certificates. Schemes such as [6] use clusterheads certificate revocation mechanism, which clusterheads are verified and monitored by their local neighbors. Every clusterhead should measure its neighbor's misbehaviors. If a clusterhead detects that a node is misbehaving, puts the corresponding certificate in his local CRL and broadcasts an accusation against the node. Any clusterhead receiving such an accusation first checks its CRL to verify that the accusation didn't originate from a node whose certificate has been revoked. If it was the accusation is ignored. Otherwise, the accused node is marked as suspect. When a threshold of legitimate accusations, i.e. K accusations is received the certificate is revoked. For handling the joining of new nodes and protecting the security services against attackers who try to compromise the administrator secret key, proactively updating secret shares is used. Schemes such as [10] use CRL approach and k or more MOCAs must agree to

revoke a certificate. Each MOCA generates a revocation certificate that contains which certificate to revoke and signs it with its key share. Then, each MOCA broadcasts the partially signed revocation certificate. Any node that collects k or more such partially signed revocation certificates can reconstruct the full revocation certificate. The list of revoked certificates or the CRL can be maintained by any node in the network since revocation certificates are not secrets but public information. The CRL can be stored at each node, the MOCAs, or at a set of specially designated nodes. Also, it is not possible to forge a revocation certificate with a valid signature on it, unless the MOCA framework is compromised. In the MOCA framework, the partial revocation certificates are distributed to all nodes in the ad hoc network via a network-wide flood, which imposes significant overhead on the network. In [8], revoking a certificate can be initiated either by m users belonging to the same cluster, or by owner of certificate. In the first case, requests need to be sent to at least k+1 server. As a result, the DCA issues a counter-certificate and adds a serial number of the revoked certificate to the global CRL. All the servers and repositories keep the CRL that contains serial numbers of revoked certificates from the entire network. CRLs are renewed by the DCA every day or more often if necessary and are broadcast to all nodes in the network. A key feature is that the CRL maintained in the repositories is timestamped, signed by the DCA, and updated every day. The corruption of the repositories is acceptable, since corrupted certificates will be detected via signature verification. If the information provided is up-to-date, it is considered correct. If it is not, another repository is accessed. Hence, the existence of t - r active repositories in each cluster ensures that the clusters operation is never interrupted. Each server participates in certificate revocation and in periodically signing the certificate revocation list (CRL) that contains the serial numbers of revoked certificates from the entire network. Once a server is compromised and detected, it cannot perform service as part of the DCA, until its share is recovered or renewed. It does not influence or affect the functioning of the cluster or the system. In schemes such as [18], a trust counter is used which depends on the behavior of node. These trust values are saved in a Neighbors' Trust Counter Table and nodes which have value lower than the threshold trust value are termed as malicious. When the node's "Expiry time" elapsed, the node broadcasts a renewal request packet to its neighbors. Node which receives a RWREQ, checks its node status from the NTT. If the nodes have value less than the *TCthr* value, then the RWREQ is dropped or else the node sends a renewal reply packet (RWREP), along with a new IT and ET field, back to the node. Due to the redundancy technique, the renewal of nodes does not consume time or halts, even if any movement of nodes or node failure or even disconnection in network occurs. Thus our work increases the integrity factor of the data in the network. Our scheme also reduces the overhead of nodes because of the redundancy factor, as it reduces the time consumption and dependency over the nodes. In [5], revoking a certificate can be initiated either by m nodes belonging to the same cluster or by a node that want to revoke its own certificate. When the revocation process is initiated by m users, their signed accusations need to be sent to at least k servers. As a result, the PDCA issues a counter-certificate and add a serial number of the revoked certificate to the global CRL. The revocation certificates are broadcast to all nodes of the network, immediately after being issued. All the server nodes and repository nodes keep the CRL that contains serial numbers of revoked certificates. When revocation is initiated by the owner of certificate, the node sends a signed request to at least k server nodes to enable the issuing of its revocation certificate. In scheme [11], the revocation is only possible when at least *k* CA nodes put their partial signatures on it. Each of the *k* CA nodes broadcasts the certificate to be revoked after putting its own signature. When the certificate to be revoked gathers *k* such partial signatures and reaches another CA node, it completes the signature, revokes the certificate, and broadcasts the revoked certificate to other CA nodes for updating their local CRLs. Any node in the network may be chosen to store the global CRL. When a node storing the global CRL has to leave the network, it needs to send the updated global CRL to an existing node in the network before it leaves. As the updating of the CRL is done by broadcast mechanism, there is a significant overhead associated with it. However, if the membership change of the nodes in the network is not very fast, the revocation will not be too frequent and the associated overhead will also be less. Choice of an optimum value of *k* is critical for efficient working of the proposed scheme. Higher the value of *k*, the more secure will be the system as more number of nodes will be involved in signature construction. However, it will also cause a large communication overhead. Thus a suitable trade-off should be made. In [15], every node maintains a CRL. If a node discovers that any other neighboring node is misbehaving, it adds that node to its CRL and floods an accusation against the node in the network. The nodes which receive this broadcast check whether the node which broadcasted this CRL is a part of its own CRL. If it is, then this broadcast is ignored, otherwise it is accepted and changes are made to the CRL.

6.Certificate Renewal

In a distributed CA, a node renews its certificate by communicating with *threshold* number of DCA nodes. For example in [15], each node sends a request for renewal to a coalition of *k* nodes. Then each DCA node checks its CRL to determine whether the old certificate has been revoked. If it has been revoked, then the nodes deny the request. Otherwise they agree to serve the request and a new partial certificate is generated and sent. In [19], requesting node first identifies a coalition of *Threshold* number of initialized nodes in its neighborhood. Then it broadcasts certificate details along with the coalition information to members of the coalition and calculate the partial certificates. If the degree of a node is small, relative to the *Threshold* value, a node may be unable to identify a coalition of initialized nodes in its neighborhood. In such cases, the requesting node will timeout and rebroadcasts the certificate renewal request. After a timeout a node might be able to identify a coalition of *Threshold* number of initialized nodes with the new set of neighbors and subsequently renew its certificate. A node may also timeout multiple times, before a successful certificate renewal. The delay experienced in certificate renewal is much higher if Average Node Degree of the network is less than the *Threshold* value. They observe that when the number of inactive nodes increases, average certificate renewal delay increase. With large delays, it becomes difficult for a node to accurately decide the time of sending renewal request. In such cases few nodes may be unable to renew their certificate on time. Average number of attempts in renewing a certificate also increases when the number of inactive nodes increases.

7.Mobility

A complete DCA system for mobile ad hoc networks must support user nodes mobility. In ad hoc network, client nodes may change their position freely or travel to other clusters, so it is desirable that user can use the DCA system even in the destination cluster or position. Thus a certificate request protocol should support user mobility and this can be done by using mobility aware routing protocol that are specially designed for mobile ad hoc network. In [6], when a clusterhead moves from his cluster, he must inform his members to elect another clusterhead or to join the other clusters. If a new clusterhead is elected, it must obtain a certificate and a partial key by requesting a coalition of K initialized clusterheads. Then, he certifies the members of his cluster using his private key. Also, when a node leaves his cluster and joins another cluster, he must obtain a new certificate from the new corresponding clusterhead. Then new clusterhead adds the generated certificate in his list and the old clusterhead deletes the old certificate from his list. In [5], a highly mobile node might leave its cluster and enter another cluster. Then, it contacts repositories of the destination cluster and sends its certificate to them and obtains their certificates. Besides that, source can send its certificate to each node it corresponds with. The certificates of the of source cluster nodes that are stored in source node are deleted, unless the mobility management protocol predicts that the node is temporarily moved to a new cluster. In this case, the node can be programmed to delete the certificates of cluster K, when it has moved to a third cluster. Another option is to keep the stored certificates if enough storage space is available. In [8], Zouridaki et al, propose a protocol to manage node migrations across clusters. When source node leaves source cluster and enters to the other cluster, it contacts any of the destination cluster repositories. Also, mobile node is introduced to the repositories of the destination cluster by sending its certificate to. Besides it can sends its certificate to each node it corresponds with. The certificates of the nodes of the source cluster that are stored in mobile node are deleted, unless the mobility management protocol predicts that the node is temporarily moved to a new cluster. In this case, the node can be programmed to delete the certificates of source cluster when it has moved to a third cluster. Another option is to keep the stored certificates if enough storage space is available.

8.Scalability

Using proper number of DCA members is very important in PDCA schemes. Because lower number of the DCA members increases the number of unsuccessful requests and more DCA members increase the DCA overheads. In [10], the number of MOCAs or *n* is determined by the characteristics of nodes, such as security or processing capability, but once it has been chosen and the system is deployed, it is expensive to change *k*. Setting *k* to a higher value has the effect of making the system more secure against possible adversaries since *k* is the number of MOCAs an adversary needs to compromise to collapse the system. But at the same time, a higher *k* value can cause more communication overhead for clients since any client needs to contact at least *k* MOCAs to get certification services. Therefore, the threshold *k* should be chosen to balance the two conflicting requirements. In this scheme they provide some guidelines for choosing an appropriate *k*, to make DCA more adaptive to varying network configurations. In [8], the number of DCA servers should be a function of the network size and the degree of resilience required against attacks. The number of servers is defined by n = 2k+1, where k is the maximum number of servers that can be compromised in a predefined period of time. In [19], one of the nodes in the network, which is called the leader node, always remains up throughout the network life time. It receives node degree values from other nodes and calculates the Average Node Degree of the network. When the number of nodes in the network reduces, leader node uses the Average Node Degree value of the network to decide the new *Threshold*. To prevent this, leader node initiates a change in the *Threshold* value of a network when the Average Node Degree falls below the current *Threshold* value. In this scheme, leader node decides to change the *Threshold* value on observing lower Average Node Degree for two consecutive time periods. Change in *Threshold* value can be as Localized or Network Wide Change in Threshold. Nodes use secure communication among the coalition members for change in *Threshold* value. Scheme such as [8] support cluster splitting and merging. If the network size increases and new clusters are formed and the number of clusters may become larger than the number of DCA servers, since the number of DCA servers is fixed n = 2k+1. In this case, not all clusters will contain one CA server. The key issue is that the CA servers are distributed into the network. If the network size decreases and some clusters are merged, the number of network clusters may become smaller than the number of DCA servers. In this case, a cluster that results from the merger of two clusters will contain the CA servers and repositories of the original clusters.

9. Security

DCA schemes establish security for wireless ad hoc network, but they can become the target of security attacks. In the MANET, threshold cryptography-based schemes are very vulnerable to the Sybil attack which is created by malicious node that impersonates many identities. Unfortunately, there is no efficient way to defeat this attack because it is difficult to bind a single identity with one node in the MANET [17]. In this section we analyze the solutions that each scheme uses to protect DCA system. For example, schemes such as [6] use encryption and authentication for prevention of attacks like eavesdropping impersonation and modification. In [4], clusterheads play key role in security related functions and they authenticate and grant affiliation status to nodes. Moreover, they are responsible for managing intercluster and intra-cluster pair-wise and group-wise keys, which are used to provide confidentiality, integrity and authentication services. Within a cluster, pairwise keys are used to secure unicast communication between nodes. Group-wise keys are used to secure group (intra-cluster) communication. Similarly, intercluster communication is protected by the use of pairwise and group-wise keys at the network backbone level, where in this case the group-wise key is shared by all clusterheads. Inter-cluster keys are managed by the clusterheads using pairwise secret master keys shared with cluster nodes. These pairwise keys are established during node affiliation, based on public key techniques. This scheme assumes that prior to joining the NTDR network; each user gets a unique *offline certificate* along with any further high-level certificates that are required to verify off-line certificates of other nodes in the network. Also if a clusterhead is evicted for security reasons, the share of the DCA private key must be considered compromised. If the evicted clusterhead is active then we must assume that the associated secret is compromised and the following actions may be done:

- One option is to accept a degree of risk and do nothing. Since k clusterheads are required to provide DCA services, the compromise of one clusterhead means at worst case that

it requires k − 1 remaining clusterheads to collude with the evicted clusterhead before the DCA private key can be obtained. In some environments this might be acceptable. It may be the case that it requires a certain number of evictions to take place before the system be no longer deemed operational.

• At the other end, the whole DCA should be bootstrapped again. This not only has implications with respect to communication cost, but potentially means that all existing online certificates would require renewal under the new DCA key pair.

It is possible to design a DCA in which DCA shareholders can be evicted by adopting a threshold scheme that offers *disenrollment* capabilities. However, such schemes typically only allow a limited number of evictions and increase the storage requirements of each clusterhead. A more acceptable compromise is probably to run the share refresh protocol. This involves a similar communication cost to bootstrapping but does not change the DCA private key. In most of the above scenarios it is still necessary to first run the promotion protocol in the event that the evicted clusterhead.

In [14], faulty servers are blacklisted and each server manages its own list. A server is inserted in such list after accusatory messages have been received from servers or clients. These messages must include a record of the false information sent by the server that is being accused. A server present in the more than *t* blacklists is excluded from the *S* in the next synchronization period and it can only be re-integrated into the DCA in special circumstances. However, accusation-based solutions are vulnerable to false accusations.

In [16] Luo et al, claim that malicious clients have no way to impersonate other nodes, because a server can verify the identity of a client by checking the ownership of the private. If the certificate update is performed frequently, an attacker is given no chance to compromise a private key before the key expires. Although exchanged messages are authenticated, it suffers from DoS attacks. However, it decreases the risk of such attacks by involving only one agent for each request.

10. Conclusion

Mobile ad hoc networks are vulnerable to numerous attacks and malicious behaviors. PKI-based security systems establish the main line of defense and protect the ad hoc network against external attackers. Digital certificates are the primary components of PKI security solutions and various methods for managing them have been defined in the literature. Distributed certificate authorities (DCA) are one of the main methods that have been used for issuing, revoking and managing the certificate in mobile ad hoc networks. They present the functionality of a certificate authority by cooperation of ad hoc network nodes and provide more fault tolerant and stable PKI services to ad hoc network users. In this paper, we analyzed the distributed certificate authority in MANETs and illustrate their properties and limitations. We specified their tasks and capabilities and demonstrate the solutions that each scheme has used for performing their tasks. All security systems incur overhead on ad hoc network, thus security solution such as distributed certificate authorities should be used according to the security threats of MANET environment and the security level which is required for user applications.

One of the problems of existing DCA systems is their inflexibility for working in low threat rate environments. On the other hand, all these schemes assume that they operate in fully hostile environment and do not adapt themselves to change in attacks and threats rates. Therefore, design of a flexible and adaptable distributed certificate authority should be considered in future researches and studies.

REFERENCES

[1] L.Zhou, F.B.Schneider, R.V.Renesse, "Coca:a secure distributed online certification authority", Journal ACM Transactions on Computer Systems (TOCS), Volume 20 Issue 4, November 2002, PP.329-368.

[2] Y. Dong, A.F.Sui, S.M.Yiu, V.O.K.Li, L.C.K. Hui, "Providing distributed certificate authority service in cluster-based mobile ad hoc networks", Computer Communications, 2007, pp. 2442–2452.

[3] D. Dhillon, T.S.Randhawa, M.Wang, L.Lamont, "Implementing a fully distributed certificate authority in an OLSR MANET ", IEEE Wireless Communications and Networking Conference, 2004, pp. 682-688.

[4] G.Chaddoud, K.Martin, "Distributed certificate authority in cluster-based ad hoc networks", 2006, Wireless Communications and Networking Conference, Vol.2, pp. 682-688.

[5] W.Rao, S.H.Xie, "Merging clustering scheme in distributed certificate authority for ad hoc network", IET International Conference on Wireless, Mobile and Multimedia Networks, 2006, pp.1-4.

[6] M.E.Elhdhili, L.B.Azzouz, F.Kamoun, "A totally distributed cluster based key management model for ad hoc networks".

[7] D.Y.Lee, H.C.Jeong, "An efficient certificate management for mobile ad-hoc network", Lecture Notes in Computer Science, 2006, Volume 4104, 355-364.

[8] C.Zouridaki, B.L.Mark, K.Gaj, R.K.Thomas, "Distributed ca-based PKI for mobile ad hoc networks using elliptic curve cryptography", First European PKI Workshop: Research and Applications, PP.232-245.

[9] P.Xia, M.Wu, K.Wang,X.Chen, "Identity-based Fully Distributed Certificate Authority in an OLSR MANET", 2008, 4th International Conference on Wireless Communications, Networking and Mobile Computing, PP.1-4.

[10] S.Yi, R.Kravets, "MOCA:Mobile Certificate Authority for Wireless Ad Hoc Networks", 2nd Annual PKI Research Workshop Program, 2004.

[11] J.Sen, M.G.Chandra, P.Balamuralidhar, Harihara S.G, H.Reddy, "A scheme of certificate authority for ad hoc networks", 2007, 18th International Workshop on Database and Expert Systems Applications, PP.615-619.

[12] M.Ge, K.Lam, "Self-initialized distributed certificate authority for mobile ad hoc network", Proceedings of the 3rd International Conference and Workshops on Advances in Information Security and Assurance, 2009, PP. 392–401.

[13] J.Kang, D.Nyang, A.Mohaisen, Y.G.Choi, "Certificate Issuing Using Proxy and Threshold Signatures in Self-initialized Ad Hoc Network", 2007, international conference on Computational science and its applications, PP.886–899.

[14] F.C.Pereira, J.D.Silva fraga, R.F.Cust´odio, "Self-adaptable and intrusion tolerant certificate authority for

mobile ad hoc networks", 2008, 22nd International Conference on Advanced Information Networking and Applications, PP. 705-712.

[15] D.Joshi, K.Namuduri, R.Pendse, "Secure, Redundant, and Fully Distributed Key Management Scheme for Mobile Ad Hoc Networks: An Analysis", 2005, Journal EURASIP Journal on Wireless Communications and Networking, PP. 579–589.

[16] J.Luo, J.P.Hubaux, P.T.Eugster, "Dictate: distributed certification authority with probabilistic freshness for ad hoc networks", 2005, IEEE Transactions on Dependable and Secure Computing, PP.311-323.

[17] H.Zhou, M.W.Mutka, L.M.Ni, "Multiple-key cryptography-based distributed certificate authority in mobile ad-hoc networks, 2005, Global Telecommunications Conference, PP 5.

[18] A.Rajaram, S.Palaniswami, "High Performance Certificate Authority Scheme in Manet", 2010, International Journal of Computer and Network Security, PP.106.

[19] S.Raghani, D.Toshniwal, R.Joshi, "Dynamic Support for Distributed Certification Authority in Mobile Ad Hoc Networks", 2006,"International Conference on Hybrid Information Technology", PP.424-432.

[20] W.E.Anderson, J.T.Michalski, B.P.V.Leeuwen, "Enhancements for Distributed Certificate Authority approaches for Mobile Wireless Ad Hoc Networks".

[21] M.Masdari , M.Maleknasab, M.Bidaki, A survey and taxonomy of name systems in mobile ad hoc networks, Journal of Network and Computer Applications, 2011.