Fully Self Organized of Certificate Authority in MANETs

Ahmad Alomari
Faculty of Mathematics and Computer Science
University of Bucharest, Romania
alomari.jordan@gmail.com

ABSTRACT: *A mobile ad hoc network (known as MANET) represents a wireless communication network, without a pre-existent subtraction or foundation and which is not reliant on any kind of centralized management. Various certificate authorities (CAs) distributed over the network, each with a periodically updated share of the secret key, is usually adopted. With this paper we hope to bring a strong contribution to making the public key management scheme more efficient especially for fully self-organized mobile ad hoc networks where the role of the dealer is played by all the nodes play. To ensure this means that each node must carry out by itself all the operations that imply nodes' public keys: initiation, distribution and revocation. The main objective of our approach is to enhance the process of building fully self certificate authority of nodes by using the Harn-line strong (n, t, n) VSS. Our proposal will provide the mobile ad hoc network flexible and efficient to make renewal and revoke certificate authority.*

Keywords: MANETs, Certificate Authority, Certificate Revocation, Certificate Renewal

Received: 17 February 2014, Revised 27 March 2014, Accepted 3 April 2014

1. Introduction

A mobile ad hoc network or MANET is a set of mobile nodes that can dynamically form an infrastructure-less based network. The networking functions are carried out by the nodes themselves in a self-organized manner meaning that the network is operated only by the end-users. For this reason, providing the security of the communication in this type of networks proves to be enormously challenging and it is realized with a big amount of effort as the number of demands of network security conflict with the demands of mobile networks, mainly due to the dynamic nature of the mobile devices. A secure networking system must have one or all of the following characteristics: confidentiality, authentication, integrity, non-repudiation and availability. When developing a security protocol for ad hoc networks certain factors need to be taken into consideration: dynamic topology, limited bandwidth and hard pressure on energy. The networks' origin and its ephemeral character, the transmission range and the nodes' capabilities are all factors that can alter the shape of a security protocol.

One of the most important issues of MANET is that of the security. Many factors must be taken into consideration before beginning to address this issue, such as the broadcasting nature of transmission, the nodes self routing environment and other factors like the open network and the mobility factor.

The authentication services already in use have their foundations in the centralized management system that is the key distribution centers or certificate authorities (CA) [1]. This kind of approach is considered to be suitable for the cases where a specific node can be protected by being reached by other nodes of the network. However, the wireless ad hoc networks are not suited for the centralized approach because it will suffer from a single-point of service denial and become inaccessible by network nodes requiring CA services. Thus a stronger CA approach is in order.

The CA mechanism is one of the security protocols proposed for ad hoc networks. The security services like authentication, integrity or security are assured for any conventional networks (like the Internet) by public key cryptography. This systems' infrastructure is composed of certificates, which provide authenticity and integrity for the public keys. Any user certificate is issued, revoked or managed by a trusted third party called certificate authority. But, due to the total absence of infrastructure in MANETs and also to the way they come together, through the dynamic collaboration between mobile and wireless devices, it is a hard job to adapt public key systems (PKI) to this kind of networks. To make that possible the tasks of certificate authority (CA) should be distributed on the user nodes or its functionality should be surpassed.

A trusted CA is usually sent in the security infrastructure to validate the authenticity of the public keys. The CA is requested to authenticate each public key before the node distributes it to the intended parties. Then the CA issues a digital certificate attaching the public key (contained in the digital certificate) to that specific node and uses its own private key to sign this digital certificate. The certificate can be authenticated by any node that contains the public key of the trusted CA. A PKI assisted by a trusted CA seems to be the most viable solution for securing MANETs. Still, due to the dynamic infrastructure of a MANET, where nodes come and go very easily, the CA functionality cannot be designated to just one node, but rather needs to be distributed in the MANET, thus avoiding the failure of the entire MANET if that specifically node decides to leave the MANET or it is compromised. To dodge this security bottleneck replicated CAs can be used. Nevertheless, this option also proves not to be scalable from administration point of view because it can create multiple points of compromise if any CA node is compromised. The ideal way to secure a MANET would be to broadcast the signing authority between a large number of nodes in such a way that multiple trusted nodes are required to band together to sign a certificate. This pattern prevails in communities wanting to protect the integrity of their membership. Initially only a few trusted members are allowed to collectively authenticate any incoming node. Newcomers are issued certificates that are valid only for a short period of time. The nodes that have recently joined a MANET are thus expected to request for renewal of their certificates very often. But, if they behave well the expiry date of the certificate will be extended and perhaps they will eventually become trustworthy enough that they will be granted the responsibility to authenticate future incoming nodes in cooperation with other trusted nodes, whereas the malicious nodes or the nodes that are observed to have broken any rule are denied renewal of their certificate which disables them from taking place in any MANET operations.

There are two categories of attacks on MANET: internal and external [2]. Internal attacks are generated by either malicious or by selfish nodes inside a network and their detection is complicated as nodes involved generate valid signatures using their private keys. Packet dropping and internal eavesdropping are two of the most common internal attacks: the nodes copy all information and speculate it without the knowledge of other nodes.

Regarding the external attacks, outsiders infiltrate the network and cause damage inside the network. Encryption and authentication are the cryptographic techniques that can prevent an external attack. As per routing, external attacks can be divided into active and passive attacks: active external attacks, like Denial of Service (DoS) attacks, packet dropping or flooding of packets, use to degrade or stop message flow between the nodes; passive external attacks are usually done by compromising the nodes and extracting vital information of the network without disrupting the network operation, which makes it basically impossible to detect, consequently making it difficult to develop security schemes for it.

We propose scheme to create and distribute the certificate authority over the node in MANETs. In our scheme every node act as a dealer which initiate sub secret share and distributed to all the nodes in MANET.

2. Preliminaries

In this section we introduce a short description of Shamir secret sharing scheme and sum of partially and fully distributed certificate authority systems.

2.1 Shamir Secret Sharing Scheme

Threshold scheme began to be implemented in 1979 by Adi Shamir who, by using polynomial interpolation ("*Lagrange interpolation*"), invented a new cryptographic method, called the secret sharing scheme. This method works as follows: we have a number of participants; every one of them has been assigned with a share of the secret; each share is combined and used to broadcast the secret between all the participants. For the secret to be restored all the shares must be reunited; each share has no use taken alone.

So we can say that in this type of scheme there are one dealer and n players. The dealer shares the secret with the players only in specific conditions. The secret can be reconstructed when any group of *t* (for threshold) or more players combine together the shares and reconstruct the secret. But a group of less than t players can not. Such a system is called a (n, t)-threshold scheme.

Shamir's scheme is easy to be proven secure: in a (n, t) scheme it can be proven that there is no difference if an attacker has $t-1$ valid shares to work with or none at all; unless he has *t* shares or more the best option to find the secret is guessing.

Sometimes a secret needs to be shared between (at least) *n* users without any $t < n$ users being able to recover the secret alone. In his work Shamir unveils the issue and provides a secret sharing scheme using polynomial interpolation as a recovery way. Namely, every user has a pair $(x_i, f(x_i), x_i \neq 0$, where $f(x)$ is a polynomial of degree *t*, and the secret is given by $f(0)$. In this configuration, one needs at least *t* shares to recover *f*, then $f(0)$. With these parameters, in order to share a secret a_0 into *d* shares, one needs to choose $t - 1$ random numbers $(a_{t-1}........a_1)$ to construct the polynomial.

$$f(x) = a_0 + a_1 x + \dots\dots\dots\dots a_{t-1} + x^{t-1}$$

After that share the sub secret between the *n* users, where $S_i = f(i)$, $i = 1,\dots..,n$, and securely transfers the share S_i to the users i; $i = 1, ..., n$.

To reconstruct the initial secret *S*, we need a subgroup at *t* least users make exchange the sub-secrets between them. Next, each user of *t* group will get *t* distinct point (i, S_i) of the polynomial. Afterwards the Lagrange interpolation is used to calculate the coefficient of the polynomial *f*

$$f(x) = \sum_{i=1}^{t} S_i \prod_{j=1\; j \neq i}^{t} \frac{x-j}{i-j}$$

Where $f(0) = a_0 = S$ this mean the shared secret may be expressed as:

$$S = \sum_{i=1}^{t} S_i \prod_{j=1\, j \neq i}^{t} \frac{-j}{i-j}$$

2.2 Partially Distributed Certificate Authority

This work is one of the first attempts that tried to deal with the issue of the key management in MANETs. It was published in Securing Ad Hoc Networks [1] and the authors, Zhou and Z. J. Haas, proposed a distributed public key management service for asynchronous ad hoc networks. This system works by allowing a set of nodes, which were assigned with the trust, to share the secret. *N* server nodes form the distributed certificate authority (DCA). Their totality benefits from a public/private key pair *K*/*k*. Each node recognizes the public key *K*. At the same time, the private key *k* is split into *n* shares $(s_1, s_2, s_3,..., s_n)$, one for every server.

Specific nodes are used to distribute the CA. Those are servers, combiners and one dealer. First two categories of nodes are employed in signing public key certificates for users. The dealer however is a particular server which knows the CA's private key. For a node joining the network the complete public key certificate is needed. This is obtained by gathering and computing all the partial signatures.

Threshold group signature is introduced at the moment the distributed certificate authority (DCA) has to sign a certificate [4]. Every node owns a share of the private key which is used to produce a partial signature. The private key shares are forwarded to a combiner *C*. Any node can accomplish this task, it only needs $t + 1$ shares so that the digital signature is restored.

Zhang et al. [5] came up with an IKM (an identity-based key management scheme). Their idea wants to be a fresh mixture of threshold and identity-based cryptography. In traditional public key management systems the distribution of certified public key is based on certificates. Their scheme tries to diminish the need for certificates, by making the public keys derive from the known identities of the mobile nodes to which some mutual data is added.

Y. Dong et al. [6] proposed a CA cluster-based architecture. They proposed two efficient schemes with low system overhead to tackle these two problems: (1) how to locate enough CA servers, and (2) how to perform the proactive share update. Compared with existing approaches, their CA architecture provides faster CA services to user nodes at reduced system overhead.

M. Omar et al. [7] proposed NetTRUST (mixed NETworks Trust infrastRUcture baSed on Threshold cryptography). Usually, in systems based on centralized trust, the single point of failure issue emerges. So, in the NetTRUST approach, this problem is tried to be diminished with a help of multiple servers that provide and distribute the services for authority certification.

2.3 Fully Distributed Models

Fully Distributed Certificate Authority Approach it was brought to our attention for the first time by Luo and Lu in [8]. Every node receives its share at the moment it joins the network by using *a* (*n*, *k*) threshold distribution scheme. It also uses verifiable and proactive secret sharing mechanisms to avoid compromising the certificate signing key and to secure the network against denial of service attacks.

It is assumed that, for establishing trust, nodes have to observe their neighbors' behavior. Also the nodes must keep a certificate revocation list (CRL) of their own. If a node is discovered as being malicious an "*accusation*" is spread throughout the network and its certificate is added to the CRL of the node that discovered it. The accused node becomes suspect and it is labeled by all the other nodes. The only way the accusation can be disregarded is that the accusatory have its certificate revoked.

N is chosen to represent all the nodes in the network which improves the practical PKI (public key infrastructure) [9], making it more available. DCA's private key SK is shared throughout all network nodes. So, if a node needs the help of the DCA it can now get in touch with any k one-hop neighbor nodes. Regarding the authentication procedure there is no difference between client and server nodes. This approach also tries to secure the certification service against the compromise coming from stronger adversaries by comprising a share update device.

A. Rachedi et al. [10] proposed a new approach to secure MANETs. The solution they offered is founded on the cluster-organized network discussed above, in which trust and CA is distributed in every cluster. The nodes with low level of trust are monitored and network security is fully self-organized. CA is chosen with the help of a clustering algorithm, which uses mobility metric and trust. In the same way each cluster gets its PKI. Still, the CA of a cluster is the cluster head, which observes and monitors the activity in each cluster.

They introduced registration authorities (RA), which are dispensable confident nodes used to enhance CAs' security in the clusters. All the certification requests pass through the RAs. They do a selection of these requests and deal with them before they send them to the CA, this operation resulting in a higher protection for the CA.

In another attempt to protect the CA and dismiss single point of attack in the clusters, a new notion, Dynamic Demilitarized Zone (DDMZ) is presented. A group of dispensable nodes compose the DDMZ. The main condition these nodes must satisfy is to be confident and to not be placed more than a hop-count away from the CA. other hierarchical routing protocols can also beneficiate from this approach.

A. Hajami, M. Elkoutbia proposed an enhanced solution for ad hoc key management based on a cauterized architecture [11]. This solution uses clusters as a framework to manage cryptographic keys in a distributed way.

The approach tries to solve key management problem in MANETs. Still, these schemes have their limitations like: congestion, administrator availability and nodes dependency etc.

To solve the problem of key management, three solutions are possible.

The first is to distribute the functions of PKI to each network node. But given the dynamics of the network, it is difficult to ensure that all members would be available.

The second solution is to designate a fixed set of nodes as permanent members of the PKI; these nodes can move freely in the network area.

The final solution is based on a clustered architecture in which the cluster-heads form the members of the PKI as will be described later.

In their work, they perform a comparative study between the second and final solution. In the following their method will be described. Their approach uses the clustering technique as well as the partially distributed PKI solution which is inspired from Threshold Secret Sharing Scheme.

They would expose the scheme in which we'd gather the cluster heads services of cluster heads in a single service called Council. Each Council node will have equal functionality and utilize the (n, k) threshold scheme for performing the cluster head functionality. The main function of this Council will be key management. A certificate will be validated by participation of at least k nodes out of n Council member. The key management cluster head function will now be able to work even when more than one (but limited to min $\{k, n - k + 1\}$) cluster head is compromised.

Dawoud and Johann proposed a key management method for MANETs that uses mobility and the routing infrastructure to manage security associations efficiently [12]. With the help of relays the keying items are broadcasted along virtual chains. Their scheme is easy to be implemented, which makes it very adequate both for motionless networks as well for low to high mobility MANETs.

2.4 Harn–Lin strong (*n, t, n*) VSS

Harn and Lin (2010) proposed a strong (n, t, n) *VSS* (Verification Secret Sharing) based on the (n, t, n) *SS* (Secret Sharing) [13].

We note that if the sum of two polynomials has degree $t - 1$ exactly, then either both polynomials have degree at most $t - 1$ or both polynomials have degree larger than $t - 1$.

2.4.1 Master secret generation phase

Each dealer p_i (shareholder) selects a random sub-polynomial $f_i(x)$ having degree $t - 1$ and the sub-secret is $S_i = f_i(0)$. The master secret is $S = \sum_{i=1}^{n} S_i$.

t

2.4.2 Master shares generation phase p_p

1. Each p_i computes sub-shares, $S_{t,n}(f_i(x)) = s_{i,1}, s_{i,2}, \ldots.., s_{i,n})$

2. Each p_i sends $S_{i,j}$ to other secretly, for $j = 1, 2, \ldots., n)$ and $i \neq j$

3. Each p_i computes the master share as $m_i = \sum_{i=1}^{n} S_{j,i}$ from n sub-shares $s_{i,j}$ for $j = 1, 2, .., n$

2.4.3 Verification phase

- Each shareholder p_i selects k random verification sub-polynomials $f_i^{\,l}(x)$, having degree $t - 1$ exactly, where $l = 1, 2\ldots, k$, and computes n verification sub-shares $S_{t,n}(f_i^{\,l}(x)) = (v_{i,j}^l, v_{i,2}^l, \ldots., v_{i,n}^l)$ for each verification sub-polynomial $f_i^{\,l}(x)$. p_i sends $v_{i,j}^l$ to other shareholder p_j secretly, for $j = 1, 2, \ldots., n$ and $j \neq i$.

- Each shareholder p_i computes verification master shares, $v_i^{\,l} = \sum_{j=1}^{n} v_{j,i}^{l}$, using its sub-shares $v_{j,i}^l$ for $j = 1, 2, \ldots, n$ and $l = 1, 2\ldots, k$. At the end of this step, each p_i has k verification master shares $V_i = \{v_i^l\}$ for $l = 1, 2\ldots, k$.

- All shareholders $\{p_i\}$, $i = 1, 2, \ldots.., n$, determine to reveal a subset G_i {say $|G_i| = k/2$ } of V_i for verification. If the degree of all $k/2$ interpolating polynomials of the revealed verification master shares is $t - 1$ exactly, the degree of interpolating polynomials of the remaining unrevealed verification master shares is also $t - 1$ exactly with very high probability.

- Each shareholder p_i releases $k/2$ values of the additive sum of the master share and each remaining unrevealed verification master share. If the degree of all the $k/2$ interpolating polynomials of released values is $t - 1$ exactly, shareholders can conclude that their master shares are generated by the polynomial $F(x)$ having $t - 1$ exactly.

2.4.4 Master secret reconstruction phase

Any master shares, $(m_{i_1}, m_{i_2}, \ldots.., m_{i_t})$, where $\{i_1, i_2, \ldots., i_t\} \in \{1, 2, \ldots.., n\}$ can reconstruct the interpolating polynomial as $R_{t,}$

$_n(m_{i_1}, m_{i_2}, \ldots.., m_{i_t}) = f_1(x) + f_2(x) + ...+ f_n(x) = F(x)$ following Lagrange interpolation formula and then obtains the master secret $S = F(0) = \sum_{i=1}^{n} S_i$.

3. Fully self Organized of Certificate Authority in MANETs by using (*n*, *t*, *n*) Secret Sharing Scheme

We consider an ad hoc wireless network with m mobile nodes. Communication between the nodes is performed via insecure channels and with a limited bandwidth. The "*m nodes*" is a dynamic number which is subject to change due to the mobile nature of the network, where the nodes come and go as they like and fail over time. Besides, m is not limited. The network can consist on a large number of nodes. The network provides neither logical infrastructure nor physical support [3].

We make the following assumptions:

1. The public key *PK* for certificate validation is well known to each node in the network.
2. Communication between multi-hop communications is considered less reliable compared with one-hop neighboring nodes.
3. Every node has at least *t* one-hop valid neighboring nodes.
4. Each node is fitted with local detection devices for the detection of malicious nodes between its neighbors.

Assume that there is a certification authority (*CA*) and *m* participant nodes in the mobile ad hoc network. *CA* will distribute a secret key to every participant node in the network. SK_{CA} secret key must beneficiate of less than *t* participants in order to function. The *CA* holds a pair of keys (PK_{CA}, SK_{CA}), PK_{CA} is the public key known by every one; SK_{CA} is the private key with external confidentiality. In our design we make extensive use of the polynomial secret sharing and fully distributed *CA* is based on an approaches described by Shamir [1] and Luo and Lu [2] respectively, and we implement our fully distributed over elliptic curve.

A random polynomial of order $t–1$ is used to share a secret, specifically the exponent of the certificate-signing key SK_{CA}, between all nodes in the network. A coalition of *t* nodes with *t* polynomial shares can potentially recover *SK* by Lagrange interpolation, while any coalition up to $t – 1$ nodes yields any information about SK_{CA}.

3.1 Initialization of scheme

In our scheme we build a fully self organized distributed certificate authority system that depends on Lin and Harn's scheme, which it depends on a (*n*, *t*, *n*) secret sharing. In most of mobile ad hoc networks (MANETs) the dealers constructs the certificate authority when the MANETs is initiated. But in our scheme the first coalition in MANET can initiate and organize the network without relying of any dealer. Every node in MANET will act as dealer to generate the master share and sub-share for all other nodes. Therefore, every node will do the same operation as all.

When the *n* nodes in MANET start to initiate the certificate authority every node will define a secret $s \in \mathbb{Z}_p$ and distribute it among them to construct the master key and we let *p* and *q* two large primes such that $q / (p – 1)$ and $g, h \in \mathbb{Z}_p$ are two elements of order *q*. We can summarize the steps as following:

- Every node will be a dealer and each nod $D_i \in \{D_1, D_2, \ldots\ldots, D_n\}$ generate sub secret polynomial $f_i(x) |= a_{i,0} + a_{i,1}x + \ldots\ldots.+ a_{i,t-1}x^{t-1}$, of degree exactly $t – 1$, in which the sub secret $a_{i,0} = f_i(0) = s_i$ and all coefficients $a_{i,0} + a_{i,1}x + \ldots\ldots.+ a_{i,t-1}$ are in $\mathbb{Z}_p$.
- Each node pick randomly $b_{i,1} + b_{i,1} \quad + \ldots\ldots.+ b_{i,t-1} \in \mathbb{Z}_p$ and generate $k_j(x)$ such that

$$k_j(x) = b_{j,1} + b_{j,2}x + \ldots\ldots.+ b_{j,t-1}x^{t-1}$$

- Each node D_i computes all sub-shares ($s_{i,j}$, $t_{i,j}$) and coefficients commitment of $f_i(x)$ and $k_j(x)$ as follows:

$$S_{i,j} = f_i(j);\; t_{i,j} = k_i(j) \;\text{ for } j = 0, 2, \ldots\ldots, n$$

- After that, every node D_i distributes the sub-share $(s_{i,j}, t_{i,j})\ \forall j = 1, 2, \ldots.., n$ and $i \neq j$ and distributes the broadcast $c_{i,v}$ to all the nodes in MANET.

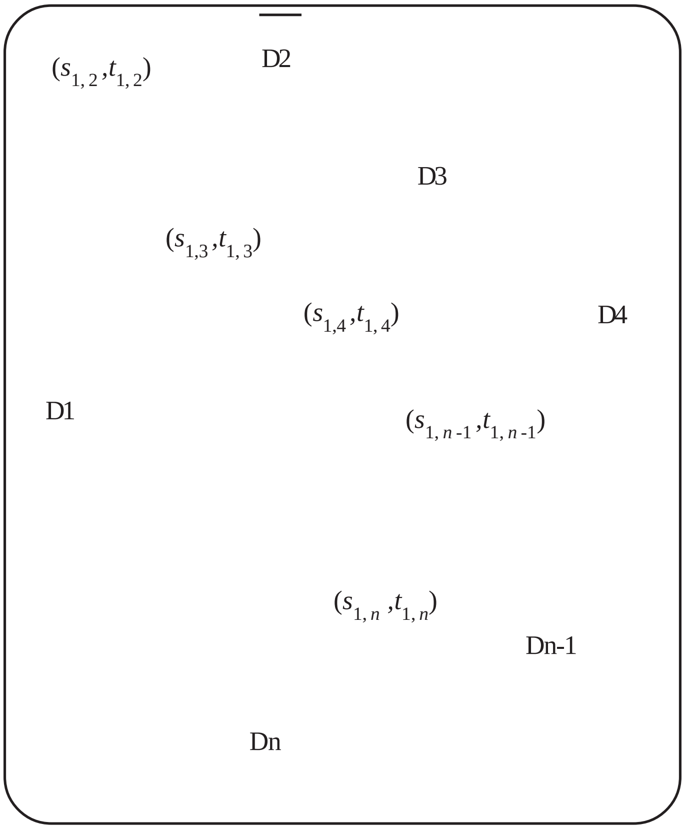

Figure 1. The node distribute sub-shares

- After D_i receives all sub-shares and broadcast information from others nodes, every node D_i computes the master share where:

$$S_i = s_{1,i} + s_{2,i} + \text{........} + s_{n,j} \text{ and } t_i = t_{1,i} + t_{2,i} + \text{........} + t_{n,j}$$

For example: node one (D_1) computes

$$S_1 = S_{1,1} + S_{2,1} + \text{............} + S_{n,1}\ (mod\ p), \text{ and } t_1 = t_{1,1} + t_{2,1} + \text{............} + t_{n,1}\ (mod\ p)$$

Node (D_2) computes

$$S_2 = S_{1,2} + S_{2,2} + \text{............} + S_{n,2}\ (mod\ p), \text{ and } t_2 = t_{1,2} + t_{2,2} + \text{............} + t_{n,2}\ (mod\ p)$$

And node (D_n) computes

$$S_n = S_{1,n} + S_{2,n} + \text{............} + S_{n,n}\ (mod\ p), \text{ and } t_n = t_{1,n} + t_{2,n} + \text{............} + t_{n,n}\ (mod\ p)$$

Also D_i computes $c_v = c_{1,v} + c_{2,v} + \text{............} + c_{n,v}\ (mod\ p)$, for $v = 0, 1, 2, \dots., t - 1$

Now every node uses Shamire's secret sharing scheme to find the master polynomial which it has the master secret

$$S_2 = S_{1,0} + S_{2,0} + \text{............} + S_{n,0}$$

Share verification: Every node D_i which has obtained master share (S_i, t_i) and all commitment values c_v for $v = 0, 1, 2, \dots.., t - 1$ can verify that all master share S_J really defines a secret by testing that

$$g^{S_i} h^{t_i} = \prod_{t-1}^{v-1} c_v^{i_j}\ (mod\ p)$$

Secret reconstruction: it is the same like in Shamire's scheme

3.2 Certificate Revocation

The issued certificate can be revoked by every node if it believes that the certificate does not posses a valid user-key. Also the node can even revoke its public key if it believes it is compromised.

In our scheme we use two certificate revocation methods: explicit and implicit.

In the explicit revocation scheme, a released certificate can be revoked if the node releases an explicit revocation statement. The revocation statement does not have to be sent to every node because each node possesses a list of nodes that request updates for the certificates that it issued. So the revocation is sent only to the nodes that regularly update it. When broadcasted, the certificate revocation arrives also to other nodes, only with a delay of the certificate exchange convergence time.

The implicit certificate revocation method is grounded on certificates' expiration time. Every certificate includes its own issuing time and a validity period (VP), which usually takes a couple of days to end. A significant operation is the right designation of VP's length, because the certificate looses its validity when this time finishes.

During the validity time of a certificate, it is presumed that a node is capable to set up communication with any node that can release certificates. Also during this period permanent exchange of certificates updates will take place and the certificate repositories of the nodes will be updated. But, if part of the certificates cannot be updated in the nodes' local repository, in the specified period of time, those certificates can be recovered with the help of the available for update certificates.

The methods described above allow nodes to be aware of the status of the certificates found in their updated certificate repositories and to be notified when other certificates are revoked. The notification presents some delay. Users are more trustworthy in the availability of the certificates when they execute authentication with the aid of key revocation. Using this mechanism also provides more confidence in the precision of the user-key bindings included in the certificates, due to their restricted validity. Also, every node can release a revocation statement as a reaction to the detection of a malicious behavior.

Key revocation is similar with the certificate revocation scheme: the public key of a node is revoked by announcing the neighbors that release certificates to it, if that node suspects its private key has been compromised. The certificate revocation mechanisms will be used to revoke the certificates that include the public key in question.

It is to be noted that the nodes are strongly encouraged to keep their certificate repositories updated. Through this method, other nodes can be assured by the authenticity of the public key. Also, the authentication of other keys can be done correctly.

The certificate revocation list (CRL) is the most common method used in the moment a node finds out that other node has been compromised. The malicious nodes' certificate is added to the CRL and an "*accusation*" against that node is broadcasted

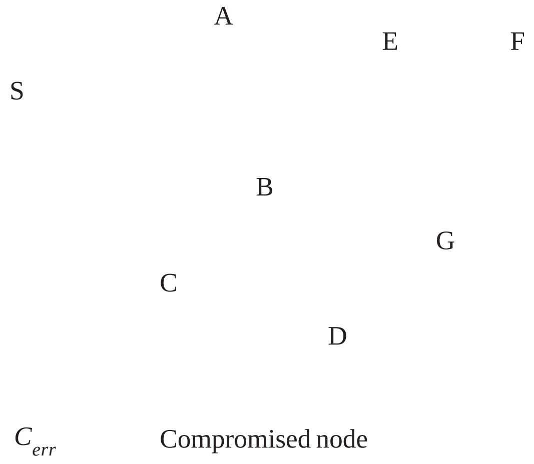

Figure 2. Broadcast certificate error

throughout the network. The nodes that receive the accusation check if it had not been issued by a node with a revoked certificate. If this is the case, the accusation is ignored. If the accusation came from a valid node, the accused node is accepted and the changes are made to the CRL. A CRL consists of a list of revoked certificates. Every node maintains a CRL.

3.3 Certificate Renewal

Certificates have an expiration date that is why their renewal is necessary. When a node D_i has to renew its certificate, a coalition of *t* neighbor nodes issues a renewal of that certificate at the request of cert. Each node in this coalition verifies the certificate proposed for renewal to not be expired or revoked. If it has been revoked, then the nodes ignore the request, otherwise the request is admitted. Every node in the coalition issues one partial certificate that contains a new expiration date and sends it back to node D_i. The partial certificates are matched by node D_i to in order to acquire cert-updated (updated certificate). In case a node becomes compromised, the partial certificate it issues is sent to the combiner. When the combiner receives this kind of certificate it will issue a certificate that is also invalid. Moreover, neighbor's nodes broadcast certificate error (C_{err}) for the compromise node (see Figure 5.4) to all nodes in MANET. The certificate of the node must be updated with the new public key. If the node changes its private and public keys, the same process occur like when the certificate is renewed.

If an adversary continuously penetrates and takes over node in an ad hoc network the best line of defense is a proactive security device, which takes care of the security of the entire network. In particular it ensures the automated recovery of the security of individual components, avoiding the use of expensive and inconvenient manual processes.

One of the important requirements of proactive secret sharing scheme is an authenticated broadcast channel and secure communication channels between the nodes. At the beginning of each time period and after initialization, when all the nodes trigger an update phase, all the nodes should perform the share renewal protocol. After triggering the share renewal protocol, each node will obtain a new share on the new $t - 1$ polynomial. The nodes should agree on the new polynomial with same secret *s* without revealing the secret.

Each node D_i where $i = 1\dots n$ randomly picks $t - 1$ numbers from the finite field.

These numbers define a polynomial $p(x)$ of degree $t - 2$.

$$p(x) = a_0 + a_1x + \dots\dots a_{t-2}x^{t-2}.$$

Each node D_i distributes the shares $p_i(x)$ among the nodes using Verifiable Secret Sharing Scheme.

Each node D_i receives all the shares $p_1(i), p_2(i), \dots\dots, p_n(i)$ and computes the new shares by adding the sum of all new shares to his old share. The new share is computed as follows:

$$y(i) = F(i) + \sum_{k=1}^{k=n} p_k(i)$$

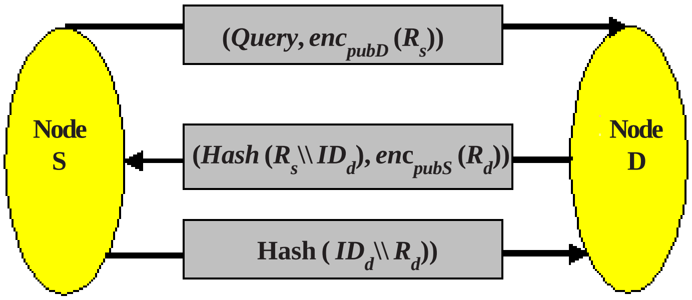

Figure 3. Mutual authentication in our scheme

where $y(i)$ is the new share of *i*-th node and $F(i)$ the old share.

In another hand When the set of nodes $(D_1, D_2,\dots\dots\dots, D_n)$ is changed to $(D_1', D_2',\dots\dots\dots, D_n')$ and the value from threshold is adjusted from t to t' that means all the MANET here need to be reconstruct. Each node D_i re-computes all sub-shares $(s_{i,j}', t_{i,j}')$ and distributes it again to the all nodes in MANET. After that they compute new master shares to produce the new master key.

3.4 Mutual Authentication in our Scheme

In this section we describe a method to enhance the security and privacy by mutual authentication in MANETs, based on random hash lock approach. We use random list stored in nodes instead of random value generator. We summarize this scheme by the following steps (see Figure 5.5):

- The source node generates random number R_s and after that encrypts it by destination public key and sends the value with request (*Query*, enc_{pubD} (R_s)) to the destination node.

- When the destination node receives the package from the source node, first decrypts it by its private key and after that hashes the identity with random number (*hash* $(R_s \backslash\backslash ID_d)$). The second step is to generate random number and encrypt it by public key of the source node enc_{pubS} (R_d)). In final the destination node sends the pair (*hash* $(R_s \backslash\backslash ID_d)$, enc_{pubS} (R_d)), containing the result of this hash operation and its own random value R_d to the source node.

- When the source node receives the pair (hash $(R_s \backslash\backslash ID_d)$, enc_{pubS} (R_d)), it will decrypt the value enc_{pubS} (R_d) by it private key to get R_d value and checks each ID_d stored in it by hashing each ID_d concatenated with R_s. Once it finds that this hash output matches the received hash result, the authentication passes the examination. After that, the source node replies to the destination node by the value formed by hashing the returned ID_d concatenated with R_d (*hash* $(ID_d \backslash\backslash R_d)$).

- When the information reaches the destination node, the node hashes its own ID_d concatenated with Rand compares it with the received value. If they are equal, the source node passes the destination to source authentication.

When all the previous steps are successfully checked then the mutual authentication between the source and destination node is done. This scheme helps the MANETs to resist many attacks such as the man in the middle attack.

3.5 A comparison of our scheme with other systems

We compare our scheme with three methods and we summarize it in Table 1:

First one is an enhanced distributed certificate authority scheme for authentication in mobile ad-hoc networks [15]. This approach attempts to improve the distributed certificate authority scheme, in order to ensure the integrity of the information and make the network safer against inside and outside attacks. This scheme uses Shamir's secret sharing scheme to which it adds a redundancy mechanism to endorse the renewal and revocation of certificates. Each hop of the traveling packet is observed and if malicious behavior exists, it is discovered with the help of various trusting devices.

This method uses three major parameters: monitoring routing cum forwarding (RCF) behavior, certificate revival and certificate revocation. Routing and forwarding packets are observed using a punish/reward system in which a trust meter increases or decreases. The intermediate nodes label the packet with its own hash value and sends further to the destination node, which examines the values of the hash function and of the trust counter. Consistent with the hash value, the counter is increased or decreased. If the trust meter drops to a value under the value set by a trust threshold, the intermediate node is labeled as malicious.

The second method is Efficient Public Key Certificate Management for Mobile Ad hoc Networks [16]. This efficient key management scheme is adequate for fully self organized mobile ad hoc networks in which the roles of the nodes are identical. The node themselves assure the services inside the network, such as: creating, storing, distributing, and revoking nodes' public keys. This scheme tries to achieve a better construction process of local certificate repositories for the nodes. The solution found is a combination of web trust concept with multipoint relay routing concept in the OLSR protocol.

The third method is providing robust and ubiquitous security support for mobile ad hoc networks [14]. Here, the authentication is distributed with the help of threshold cryptography and shared secrets. The major interest of this type of scheme is to share a secret key k among an arbitrarily large community using a secret polynomial $f(x)$. If the degree of $f(x)$ is $(k-1)$, any k members of the community can recover the secret key, while any members less than *k* reveals no information of the secret.

As shown our proposed protocol provides more security and authentication between nodes, improving also the security of the network because we use mutual authentication in MANETs, based on random hash lock approach, we use random list stored in nodes instead random value generator.

4. Security Analysis

In our scheme we use distributed certificate authority which it depends on secret sharing scheme. Most models of certificate authority in MANETs use a dealer to share a secret between n participants (*excluding the dealer*). The way the secret is divided makes it possible for only a group of the whole participants to restore it. Nevertheless, it is possible that the participants might not be capable to retrieve the secret if the dealer or other participants conduct a malicious behavior. This kind of conduct can be hindered through the implementation of a security protocol which allows the share recipients' to check most of the dealing. This approach can work if all the participants (*including the dealer*) are honest.

In verifiable secret sharing (VSS) [13] the main objective is to withstand the misbehavior of the nodes. Some of this conduct includes dishonest transmission of shares to one, some or all the player, which submit those shares when the reconstruction process takes place. To use VSS it is required to maintain available the private channels between the node and each participant individually. Still, it is clear that the communication through private channels cannot be verified in public.

In our scheme every node behaves as a dealer and wants to assist in creating and sharing of a master secret. Each node chooses a random secret, calling it a sub-secret. If Shamir's share generation algorithm is employed, the sub-secret can be shared between the nodes with the help of sub-shares. The master share can be created by combining every sub-share of each participant so that in final, the master secret could be reconstructed. To do this Shamir's secret reconstruction algorithm is employed, using any t or more than t master shares.

It is easy to observe that the above (n, t, n) *SS* uses Shamir's (t, n) SS as building block and is based on the additive homomorphism property. Since Shamir's (t, n) SS is information-theoretically secure, this (n, t, n) SS is also information-theoretically secure. In addition, the sizes of every master share and of every share in Shamir's (t, n) SS are identical. The same approach can be applied on any linear (t, n) SS to convert any (t, n) SS into an efficient (n, t, n) SS.

A verifiable secret sharing scheme allows all nodes that own shares to combine their efforts in checking if their shares are *t*-consistent. In a secret sharing scheme that engages many dealers the most wanted achievement is a controllable environment, with mutual verification, because the dealers usually do not trust each other. In the (n, t, n) SS, the master share of each shareholder is a combination of *n* sub-shares generated by *n* mutually distrusted dealers. Thus, verifiability of these master shares is very important.

4.1 Backward Secrecy

When a node quits the network, it should not be capable to decrypt the future encrypted passing. In proposed certificate authority scheme, every time a node departs, collision nodes regenerate sub secret sharing and distribute it in the group and after that generate master polynomial which it used to produce the new master secret sharing. Using this method, the key update is secure and backward secrecy is kept in the network.

4.2 Forward Secrecy

Forward secrecy says that it should not be possible for the past encrypted passing to be decrypted by a new node that joins the network. On joining of new node, every node in the mobile ad hoc network compute $s_{i,n+1} = f_i(n+1)$, $t_{i,n+1} = k_i(n+1)$ and send it to the new node after that the new node can compute the master secret share by using Shamir's scheme, ensuring forward secrecy.

4.3 Mutual Authentication

In proposed certificate authority in mobile ad hoc network, both new node and group nodes in MANETs authenticate each other mutually, at the time of network joining. If the authentication is a success, node can join the network. When two nodes want to

Requirements	Efficient public key certificate management for mobile ad hoc networks	Providing Robust and Ubiquitous Security Support for Mobile Ad hoc Networks – Kong	An Enhanced Distributed Certificate Authority Scheme for Authentication in Mobile Ad-hoc Networks	Our scheme (Fully self organized of Certificate Authority in MANETs by using (*n*, *t*, *n*) Secret Sharing Scheme)
1. Distributed authentication	Fully distributed certification method. Every node acts as a CA.	Totally distributed. Scales well to large networks	Totally distributed. Scales well to large networks	Totally distributed. Scales well to large networks
2. Resource awareness	Each node maintains two certificate repositories, the updated and the non-updated repositories	The generation and distribution of keys using complex polynomial functions uses much time and resources.	Using Sharing Scheme with Redundancy to reconstruct certificate authority increases the data integrity and helps the network nodes to be more mobile.	Each node maintains two certificate repositories, the updated and non-updated repositories
3. (a) Creation CA	The nodes themselves carry out the certification through chains of certificates.	Requires at least k neighbors which might be a bottleneck	The CA key is shared to a set of nodes N which depend on Sharing Scheme with Redundancy: the reconstruction is possible even if the minimum set of necessary key (k) is not available during reconstruction.	Uses self–signed certificates, hence more robust than a shared key based mechanism
3. (b) Renewal CA	Certificate renewal may be developed on demand. A node that receives an expired certificate in its repository must ask for a renewal of that certificate.	Same as issuance	Modified Shamir's secret sharing with Redundancy: each node is being introduced with a witness.	Using proactive certificate method and same as issuance
3. (c) Revocation CA	Explicit revocation causes delay between far-away nodes in the network.	System CRL table stored at each node and hence memory intensive.	Increase and decrease a trust counter depending on the behavior of the node. These trust values are saved in Neighbor's Trust Counter Table (NTT)	Using two certificate revocation schemes: explicit and implicit and also use CRL system to store the revocation nodes
4. Dealer	Exist dealer initiate the MANET, every node have own certificate authority	Exist dealer initiate the MANET	Exist dealer initiate the MANET	Every node act as dealer.
5. Mutual Authentication	Does not implement	Does not implement	Does not implement	Implemented between the source and the destination node

Table 1. Comparison our scheme with others methodologies

communicate inside the network, they authenticate mutually by sending each other’s Digital Signature.

4.4 Man in Middle Attack

Man in the Middle attack is a kind of active attack in which the adversary maintains its invisibility between two nodes, like source and destination nodes. Attacker splits the connection in two: one between node *S* and the attacker and the second between the attacker and node *D*. The two nodes, *S* and *D*, think that they are communicating with each other, while they communicate with the attacker located between them.

Most of the schemes are vulnerable to Man in the Middle attack. For example, a node sends to a new joining node its public key. In response of the request, the node creates a session key and sends it to the new joining node, encrypted with the new joining node public key. In this scheme, an attacker may exist between new joining node and the node of MANET; attacker can seize the public key of the new node and send its public key to the node in MANET. Then the node in MANET shares the session key with the attacker, which, in its turn shares the session key with new joining node. But in proposed certificate authority system, both new node and any node in MANET authenticate each other using challenge-response protocol. Hence, our certificate authority system is not vulnerable to Man in the Middle attack.

5. Conclusion

In this paper, we proposed fully self organized of certificate authority in MANETs by using (*n*, *t*, *n*) secret sharing scheme which it depend on Harn-line strong (*n*, *t*, *n*) VSS. The goal of the presented method is the improvement in the process of building fully self certificate authority of nodes. Our proposal will provide the mobile ad hoc network flexible and efficient to make renewal and revoke certificate authority. Proposed certificate authority is a decentralized scheme combining both symmetric and asymmetric cryptographic algorithms; which maintains forward and backward secrecy and provides security against many attacks such as reply attack, man in the middle attack etc. The node in our scheme acts as the dealer which can generate the certificate authority with out need to the third party or main dealer.

References

[1] Yi, S., Kravets, R. (2001). Practical PKI for Ad Hoc Wireless Networks, Department of Computer Science, University of Illinois, Technical Report UIUCDCS-R-2002-2273, UILU-ENG-2002-1717, August.

[2] Marco Conti. (2003). Body, Personal and Local Ad Hoc Wireless Networks, in Book The Handbook of Ad Hoc Wireless Networks (Chapter 1), CRC Press LLC.

[3] Shamir, A. (1979). How to Share a Secret, *Comm. ACM*, 22, p. 612-613.

[4] Zhou, L., Haas, Z. J. (1999). Securing Ad Hoc Networks. *IEEE Networks*, 13 (6).

[5] Zhang, Y., Liu, W., Lou, W., Fang, Y. (2006). Securing Mobile ad hoc Networks with Certificateless Public Keys. IEEE Transactions on Dependable and Secure Computing.

[6] Dong, Y., Sui, A. F., Yiu, S., Li, V. O., Hui. L. C. (2007). Providing Distributed Certificate Authority Service in Cluster-based Mobile ad hoc Networks. Elsevier, Computer Communications, May.

[7] Omar, M., Challal, Y., Bouabdallah, A. (2007). Nettrust: mixed Networks Trust Infrastructure based on Threshold Cryptography. *In*: Proceedings of IEEE Securecom’07/SECOVAL, September.

[8] Luo, H., Lu, S. (2000). Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks., Technical Report200030, UCLA Computer Science Department [5] J-P. Hubaux, L. Buttyán and S. Capkun.

[9] Yi, S., Kravets, R. (2001). Practical PKI for Ad Hoc Wireless Networks, Department of Computer Science, University of Illinois, Technical Report UIUCDCS-R-2002-2273, UILU-ENG-2002-1717, August.

[10] Rachedi., Benslimane. A. (2006). Trust and Mobility-based Clustering Algorithm for Secure Mobile ad hoc Networks. *International Conference on Systems and Networks Communication*.

[11] Hajami, A., Elkoutbia, M. (2012). Distributed Key Management Scheme for MANET using Council Architecture SI2M, ENSIAS, Université Mohammed V – Souissi B.P. 715, ENSIAS – Rabat Maroc.

[12] Dawoud, S., Johann van der. (2012). Fully Distributed Authority-Based Key Management for Mobile Ad Hoc Networks Dean of Engineering, Victoria University, Uganda. dawoud.shenouda@vu.ac.ug.

[13] Lein Harna, Changlu Lin, Strong (*n*, *t*, *n*) Verifiable Secret Sharing Scheme, *Information Sciences*, 180 (2010) 3059–3064

[14] Rajaram Ayyasamy, Palaniswami Subramani. An Enhanced Distributed Certificate AuthorityScheme for Authentication in Mobile Ad-hoc Networks, *The International Arab Journal of Information Technology*, 9 (3), May.

[15] Caballero-Gil, P., Hern´andez-Goya. C. (2011). Efficient Public Key Certificate Management for Mobile Ad Hoc Networks *EURASIP Journal onWireless Communications and Networking*.

[16] Luo, H., Lu, S. (2005). Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks., Technical Report 200030, UCLA Computer Science Department [5] J-P. Hubaux, L. Buttyán and S. Capkun.