A Council-based Distributed Key Management Scheme for MANETs

Abdelmajid HAJAMI[#], Mohammed ELKOUTBI[#]

[#]*SI2M, ENSIAS, Université Mohammed V Souissi*
B.P. 715, ENSIAS Rabat Morocco

abdelmajid_hajami@yahoo.fr
elkoutbi@ensias.ma

***Abstract*—Mobile ad hoc networks (MANETs) have been proposed as an extremely flexible technology for establishing wireless communications. In comparison with fixed networks, some new security issues have arisen with the introduction of MANETs. Secure routing, in particular, is an important and complicated issue. Clustering is commonly used in order to limit the amount of secure routing information. In this work, we propose an enhanced solution for ad hoc key management based on a cauterized architecture. This solution uses clusters as a framework to manage cryptographic keys in a distributed way. This paper sheds light on the key management algorithm for the OLSR protocol standard. Our algorithm takes into account the node mobility and engenders major improvements regarding the number of elected cluster heads to create a PKI council. Our objective is to distribute the certification authority functions for a reduced and less mobile cluster heads that will serve for keys exchange.**

***Keywords*- Key Management; MANET; Clustering.**

I. INTRODUCTION *(HEADING 1)*

In Mobile Ad hoc Networks, devices may have different configurations, and must cooperate to ensure the existence of such networks. MANET devices are free to move in the network, re-enter and leave at will, which shows the spontaneous nature of this type of networks. In addition, these networks do not support the existence of any supervisory or management authority, which provides equipments the same roles in the functioning of the network.

To ensure communication between network devices, MANETs use the radio link. This allows a malicious node to infiltrate easily to disrupt the network. To prevent such behavior, a cryptographic authentification system should be established. However, the authentification system should include a trusted entity that will manage the cryptographic keys.

Effective management of keys, or digital certificates holding the keys, is one of the key factors for the successful wide-spread deployment of cryptographic keys. PKI (Public Key Infrastructure), an infrastructure for managing digital certificates, was introduced for this purpose. The most important component of PKI is the CA (Certificate Authority), the trusted entity in the system that vouches for the validity of digital certificates. The success of PKI depends on the availability of the CA to the nodes in the network since a node must correspond with the CA to get a certificate, check the status of another node's certificate, acquire another node's digital certificate, and so on.

However, connectivity, which was assumed to be good in previous wired PKI solutions, is no longer stable in ad hoc networks. Unfortunately, maintaining connectivity is one of the main challenges in ad hoc networks, since the inherent infrastructurelessness of ad hoc networks makes it hard to guarantee any kind of connectivity. Another serious problem inherent in ad hoc networks is the physical vulnerability of the nodes themselves. Considering that most ad hoc networks will be deployed with mobile nodes, the possibility of the nodes being captured or compromised is higher in wired networks than in those with stationary hosts. With an infrastructure-based solution, mobile nodes may store all sensitive information in the infrastructure and maintain minimal information in the device. Since there is no stable entity in an ad hoc network, the vulnerability of its nodes is even higher.

Our proposed solutions to provide PKI for ad hoc networks deal with the physical vulnerability of the nodes by employing the distribution of CA functionality across multiple nodes and using cryptography threshold. This approach also increases the availability of the CA.

In this work, we will present a solution for managing cryptographic keys based on a clustered architecture for securing the OLSR routing protocol.

Our solution describes how to build key- management infrastructure on a clustered architecture in which a set of nodes in the network are selected using a specific criterion to represent the other nodes in the network [1][2]. These elected nodes, which are cluster-heads of the network, will form what we call the council of the PKI.

This paper is organized as follows: in Part II, we'll present an overview of the OLSR standard protocol. Part III will present an overview of the key management in ad hoc networks.

In Part IV we'll give an overview of the clustering solution that we have adopted. In Part V, we'll discuss in more detail our key management proposal in which we will show the results obtained from the performed simulations. Finally, in Part VI of this paper, we would put in test the robustness of our PKI solution by applying two types of attacks on our proposed architecture.

II. The OLSR Protocol

The optimized link state routing (OLSR) protocol [3] is a proactive routing protocol that employs an efficient link state packet forwarding a mechanism called multipoint relaying. Optimizations are done in two ways: by reducing the size of the control packets and also by reducing the number of links that are used for forwarding the link state packages. The reduction in the size of link state packets is made by declaring only a subset of the links in the link state updates. The subset neighbors that are designated for link state updates are assigned the responsibility of packet forwarding are called multipoint relays. The optimization by the use of multipoint relaying facilitates periodic link state updates. The link state update mechanism does not generate any other control packet when a link breaks or when a link is newly added. The link state update optimization achieves higher efficiency when operating in highly dense networks. The set consisting of nodes that are multipoint relays is referred to as MPRset. Each given node in the network pinpoints an MPRset that processes and forwards every link state packet that this node originates. Each node maintains a subset of neighbors called MPR selectors, which is nothing than the set of neighbors that have selected the node as a multipoint relay. A node forwards packets that are received from nodes belonging to its MPRSelector set. The members of both MPRset and MPRSelectors keep changing over time. The members of the MPRset of a node are selected in such a manner that every node in the node's two-hop neighborhood has a bidirectional link with the node.

The selection of nodes that constitute the MPRset significantly affects the performance of OLSR. In order to decide on the membership of the nodes in the MPRset, a node periodically sends Hello messages that contain the list of neighbors with which the node has a bidirectional link. The nodes that receive this Hello packet update their own two-hop topology table.

The selection of multipoint relays is also indicated in the Hello packet. A data structure called neighbor table is used to store the list of neighbors, the two-hop neighbors, and the status of neighbor nodes. The neighbor nodes can be in one of the three possible link status states, that is, unidirectional, bidirectional, and multipoint relay.

III. Overviewing The Key Management In MANETs

The aim of this section is to show some solutions for key management in ad hoc networks. The major problem in providing security services in such infrastructure with few networks is how to manage the cryptographic keys that are needed. In order to design practical and sufficient key management systems it is necessary to understand the characteristics of ad hoc networks and why traditional key management systems cannot be used [4].

A. *Key Management*

As in any distributed system, in ad hoc networks the security is based on the use of a proper key management system. As ad hoc networks significantly vary from each other in many aspects, an environment-specific and efficient key management system is needed.

The security in networking depends, in many cases, on proper key management. Key management consists of various services, each of which is vital for the security of the networking systems. The services must account for these issues: Trust model, Cryptosystems, Key creation, Key storage and Key distribution [5].

The key management service must ensure that the generated keys are securely distributed to their owners. Any key that has to be kept secret must be distributed so that confidentiality, authenticity and integrity are not violated. For instance whenever symmetric keys are applied, both or all of the parties involved must receive the key securely. In public-key cryptography the key distribution mechanism must guarantee that private keys are delivered only to authorized parties. The distribution of public keys need not preserve confidentiality, but the integrity and authenticity of the keys must still be ensured. We showed several solutions for key management in ad hoc networks.

1) *Partially Distributed Certificate Authority*

This solution proposed by Zhou and Hass [6] uses a (k, n) threshold scheme to distribute the services of the certificate authority to a set of specialized server nodes. Each of these nodes is capable of generating a partial certificate using their share of the certificate signing key skCA, but only by combining k such partial certificates can a valid certificate be obtained. The solution is suitable for planned, long-term ad hoc networks. Since it is based on public key encryption it requires that all the nodes are capable of performing the necessary computations. Finally it is assumed that subsets of the nodes are willing or able to take on the specialized server role.

This solution requires that a server and organizational/administrative infrastructure is available and therefore is only applicable to a subset of ad hoc network applications.

Viewed from a functional standpoint the solution has a number of faults or weaknesses of which the lack of a certificate revocation mechanism is the most critical. Any certificate-based solution should, considering the risk of compromise in ad hoc networks, provide such a mechanism.

Also the solution requires that the server nodes store all of the issued certificates. This requires a synchronization mechanism that propagates any new certificates to all the servers. It must also handle the case when the network has been segmented and later re-joined.

2) *Fully Distributed Certificate Authority*

This solution is first described by Luo and Lu in [7]. It uses a (k, n) threshold scheme to distribute an RSA signing certificate key to all nodes in the network. It also uses verifiable and proactive secret sharing mechanisms to protect against denial of service attacks and compromise of the signing certificate key. This solution is aimed towards planned, long-term ad hoc networks with nodes capable of public key encryption. However, since the service is distributed among all the nodes when they join the network, there is no need in electing or choosing any specialized server nodes.

Similar to the partially distributed Certificate Authority (CA) this solution requires an organizational/administrative infrastructure to provide the registration and initialization services. The main benefit of this solution is its availability and that , unlike the other certificate based solution proposed, provides a certificate revocation mechanism.

Since all nodes are part of the CA service, it is sufficient that a requesting node has k one-hop neighbors for the CA service to be available. The amount of the network traffic width is also limited.

The cost of achieving this high availability is a set of rather complex maintenance protocols, e.g. the share initialization and the share update protocols. A larger number of shares is also exposed to compromise since each node has its own share as compared to only the specialized server nodes in the partially distributed solution. The k parameter therefore need to be larger since an attacker may be able to compromise a larger number of shares between each share update. This in turn affects the availability of the service. The solution must also provide for a synchronization mechanism in the case of network segmentations.

The proposed certificate revocation method assumes that each node is capable of monitoring the behavior of all its one-hop neighbors. This assumption, however, may be too strong in certain ad hoc networks.

3) Self Issued Certificates

This solution is proposed by Hubaux [8] and provides a public key management solution similar to PGP (Pretty Good Privacy) in the sense that certificates are issued by the users themselves without the involvement of any certification authority. Unlike the public key based solutions, this one is intended to function in spontaneous ad hoc networks where the nodes do not have any prior relationship. Nevertheless, due to this, it requires an initial phase during which its effectiveness is limited and therefore it is unsuitable for short-term networks. Since it is based on public key encryption it requires that the nodes have sufficient computational capacity.

The main benefit of this solution is that it does not require any form of infrastructure neither routing, server or organizational/administrative. However it lacks a certificate revocation mechanism. Also like PGP it has problems during its initial stages before the number of certificates issued reaches a critical amount. This solution also assumes the PGP terminology being called trusted introducers or even meta-introducers. A trusted introducer is a user that is trusted to introduce other users, i.e. to issue certificates to other users. A meta-introducer is a trusted introducer that is trusted to introduce other trusted introducers. [9]

4) Secure Pebblenets

This solution proposed by Basagni [10] provides a distributed key management system based on symmetric encryption. The solution provides group authentification, message integrity and confidentiality.

This solution is suitable for planned and distributed, long-term ad hoc networks consisting of low performance nodes that are unable to perform public key encryption

This solution based on symmetric cryptography requires an organizational/administrative infrastructure that initializes the network nodes with the shared group identity key and additional other parameters. The main weakness of this solution is that it requires that the nodes maintain a tamper-resistant storage. Such a requirement excludes the use of standard networking devices since these typically don't include any tamper-resistant memory. If the group identity key is compromised then all the network nodes need to be re-initialized with a new group identity key.

Finally since only group authentification is supported this solution is not applicable in applications where the communication is peer-to-peer.

5) Demonstrative Identification

This solution proposed by Balfanz [11] presents a mechanism for trust relationships in local ad hoc networks where the network nodes have no prior relationship with each other.

Examples of such local ad hoc networks could be a group of people at a meeting wishing to setup a temporary network or a PDA wishing to temporarily connect to a printer. Since the solution does not require that the nodes have any prior relationship, it is suitable for spontaneous, localized ad hoc networks. It is unsuitable for distributed ad hoc networks since it requires that the nodes be in a close proximity of each other during the initial bootstrapping. It allows the participating nodes to have diverse capabilities, i.e. some are limited to symmetric encryption while others are capable of public key encryption.

All previous solutions have required either an organizational/administrative infrastructure or some sort of social interaction as in the solution based on self issued certificates. The use of demonstrative identification however allows the formation of a secure ad hoc network in a purely self-configured way. As an example two users need only to point their PDAs towards each other. The PDAs then automatically exchange the authentification information required to secure the following communications.

A possible down-side is that the networking devices must be equipped with some sort of location-limited channel. However since the majority of portable devices, e.g. PDAs and laptops are equipped with an infrared interface this should not be a problem. Also this solution is only applicable for localized ad hoc networks.

IV. The Clustering Solution

The network can be considered as a set of areas (or clusters). Each cluster is formed around a representative called Cluster Head. Cluster Heads are selected according to a well defined criterion.

A cluster is designated by an identifier that relates to its representative (i.e. its cluster head). Each node in the network carries the cluster identifier to which it belongs.

Our proposal presents a simple, light and quiet solution [1][2]. First, our proposal does not add any new control message and the network is not overloaded or slowed at all. No changes are made to standard control messages. Our solution works transparently with the OLSR standard protocol. Clusters are formed around the nodes with the densest environment; in other words, the node that has the largest number of symmetric neighbors is selected as the cluster head.

In this way, we are sure that the cluster is represented by the node that covers the largest number of nodes in the cluster.

V. Key Management Scheme

As in any distributed system, in ad hoc networks the security is based on the use of a proper key management system. As ad hoc networks significantly vary from each other in many aspects, an environment-specific and efficient key management system is needed.

The security in networking depends, in many cases, on proper key management. Key management consists of various services, each of which is vital for the security of the networking systems. The services must provide solutions to be able to answer the following questions: Trust model, Cryptosystems, Key creation, Key storage and Key distribution [12].

VI. The Proposed Solution

As described previously, the approaches presented in the literature tried to solve key management problem in ad hoc networks, but these solutions still carry many limits (administrator availability and congestion, dependence of nodes on the administrator and so on). To solve the problem of key management, three solutions are possible. The first is to distribute the functions of PKI on all nodes in the network. But given the dynamics of the network, it is difficult to ensure that all members be available. The second solution is to designate a fixed set of nodes as permanent members of the PKI; these nodes are free to move 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 our work, we perform a comparative study between the second and final solution. In this section, we are going to describe the approach that we propose for key management in ad hoc networks, which is based on both the clustering technique and the partially distributed PKI solution which is inspired from Threshold Secret Sharing Scheme.

- (k, n) Threshold Secret Sharing Scheme

In secret sharing scheme, a secret is shared among a group of users called shareholders. The secret is shared in such a way that no single user can deduce the secret from his share alone and in order to construct the secret, one need to combine a sufficient number of shares. Adi Shamir [13] proposed a classical (k,n) secret sharing algorithm based on polynomial interpolation. The scheme describes how a secret S can be divided in to n partial shares (S1,S2,...,Sn) where a minimum of k out of n are partial shares are needed to generate a secret S. The threshold value k is a balancing point between fault tolerance and service availability. Asmuth and Bloom [14], Brickell [15], and Karin-Greene-Hellman [16] have enhanced this work. Also, work has been done in the issues related to verifiable secret sharing [17] and verifiable secret redistribution [18].

A. Description of the scheme

In this section, and once clusters are formed and heads are designated, as described in [1][2], we 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 (k,n) 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.

In our scheme, we propose a novel architecture that we call 'Council'- based clusters. The scheme uses a collaborative approach to perform as Council-based clusters throughout the network, making it as extremely efficient as possible. Once the Council- based clusters are formed, each Council member can apply (k,n) threshold scheme in a way that a minimum of k cluster heads out of n need to participate together to perform any CA function. For example, for key distribution functionality, Every Council member (each serving as CA) will serve his cluster members. By having multi-cluster heads, the network will be able to work even when more than one (but limited to min {k,n-k+1}) cluster heads are compromised.

- Key Management Scheme on Council- Based Cluster

Key management is an important aspect of ad hoc network security. To ensure security using public key cryptography scheme, each node carries a public-private key pair and a certificate issued by the CA. As discussed earlier, one of the cluster head functionalities can be to function as the CA. A CA certifies that a public key belongs to a particular entity.

Having a single centralized CA is not suitable for highly vulnerable ad hoc networks. Using our scheme, the process of certification can be distributed among all Council nodes within each cluster. We divided our study into two major parties. In the first part, the council is composed of members designated in advance and do not change during the lifetime of the network. This is what we called fixed membership architecture. And in the second part, council members are formed by the heads of clusters; that is what we called clustered architecture. Council issues a certificate to a member node‘s public key by digitally signing it with the private key of the cluster. In order to validate the certificate of a node, at least k Council members out of the n need to work together and combine their contributions. Since at least k among n contributions are required to validate the node’s certificate,the system will work even if more than one, but limited to min (k,n-k+1),Council members are compromised.

- Why Limited to Min (k, n-k+1) Compromised the Cluster Heads

In the above section we have mentioned that the cluster head functionality will be able to work even when more than one but limited to min {k,n-k+1}cluster heads are compromised. Let us discuss why our (k,n)threshold scheme is limited to min {k,n-k+1}. In (k,n)secret sharing scheme, a minimum of k cluster heads out of n need to participate together to perform any cluster head functionality. If k or more cluster heads are compromised they will be able to combine their secret share together to perform any compromised cluster head functionality. Thus the total number of compromised nodes cannot exceed k-1. What is more is that in order to perform cluster head service the operation will require at least k non-compromising cluster heads; the system will not if the number of compromised cluster heads are equal to or greater than n-k+1. In general our (k, n) secret sharing scheme will work for any T compromised cluster heads where 1< T < min {k,n-k+1}. For ex. in (5, 12) secret scheme, the system will not work for 5 or more compromised cluster heads as minimum of 5 compromised cluster heads can participate together to perform any cluster head functionality. The (7,12)scheme will not work if 6 or more cluster heads are compromised, as a minimum of 7 cluster heads are required for making the decision.

- Finding (k, n)

We have also addressed the problem of choosing a suitable (k,n) pair on Council based clusters. Not being uniformly distributed, the whole network makes the choice of (k,n) difficult. We find the value of n in an adaptive fashion depending on the availability in the networks. In short, the number of Council members per cluster will give us the value of n. The threshold value k is a balancing point between fault tolerance and service availability. Let’s discuss the special cases of choosing k:

• k =1: The secret is shared by n nodes and anyone of them can get the secret using just 1 share. This scheme is similar to a single cluster head and hence vulnerable to a single point of failure.

• k =n: The secret is shared by n nodes and all these nodes need to participate together with their shares in order to get the secret. This scheme provides maximum security but requires accessibility to all the nodes. For highly secure networks like military applications, we will choose k =n and apply (n,n) threshold secret share concept on Council.

• 1<k <n: We chose such a k in a way that there should be a balance between security and availability.

- Scheme steps

The scheme can be explained by the following steps:
1. startup scenario: when starting the network, at least k nodes among members, must share in face-to-face a physical starting key. This key will serve as a session key that will be changed immediately after the start of the network. In this way, any unwanted intrusion will be rejected. Nodes that create the network for the first time are permanent members of the council of PKI. They have a maximum trust metric, and take care of authenticating other nodes that join the network later.
2. After starting the network, if a node arrives for the first time, it must contact in face-to-face one of the permanent members to have a physical certificate. This certificate contains a session key that will enable the new node to connect to the network.
During network operation, each PKI council member records all delivered certificate and broadcast it to the rest of the council. Each network node that is not part of the council must register all obtained certificates.

If a node leaves and joins the network, or if it changes the cluster due to a move, it must be authenticated by one of the council members (as used architecture: fixed membership architecture or clustered architecture) by presenting its first physical certificate. Based on this certificate, the council member broadcasts a request for certificate validation by the other council members. If the authenticator member receives at least k positive responses among n, the node that wants to authenticate will be accepted and the certificate will be delivered.

B. *Performance analysis*

To see the pertinence of this approach and to measure the effect that will cause the implementation of our algorithm in an OLSR network, we performed several simulations with a variable number of nodes and different nodes velocity.

We used NS2 [20] as network simulator with the following parameters:

Parameter	**value**
Simulation area	1000 x 1000
Radio range	250 m
Number of nodes	From 10 to 100 by step of 10
Velocity of nodes	From 0 m/s to 40 m/s by step of 5
Simulation time	300 s

We separated our operations into two phases. In the first phase, we decided to measure the impact of our PKI solution on network performance. Parameters that we evaluated are:

the end to end delay, the average through put, the packet delivery fraction (PDF), normalized routing load generated (NRL), the number of collisions in the network, routing loops in the network, the rate of traffic and the number of non-route in the network. In this phase, we made a comparison for three different architectures: an OLSR network without a PKI, an OLSR network with permanent PKI members and an OLSR network with cluster -based PKI.

In the second phase, we defined a set of performance metrics to evaluate the effectiveness of our key management solution. These parameters are: the delivery delay of a certificate representing the time elapsed since the request to delivery of a certificate, the CDF (Certificate Delivery Fraction) which represents the percentage of certificates issued and finally the response time of the PKI which represents the time elapsed between the start of the network and the delivery of the first certificate. We also measure the influence of the threshold parameters (k and n) in order to observe the behavior of the performance of the PKI.

The first phase results are as follows.

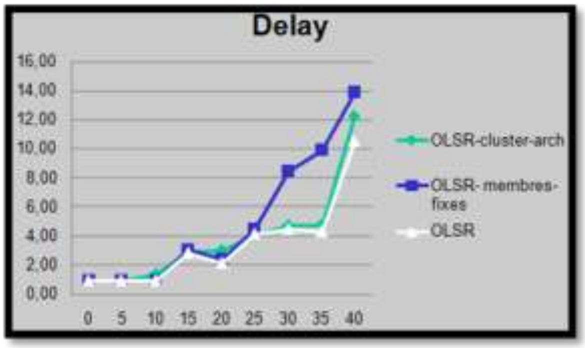

Figure 1- end to end delay in term of node's velocity

Figure 1 shows the end to end delay depending on the speed of the nodes in the network. For low speeds, we notice that the three architectures behave in the same way, while for speeds above 30 m/s, the architecture with permanent members creates an additional delay due to the nodes high mobility. Change of the number of nodes in the network has no effect on the delay parameter.

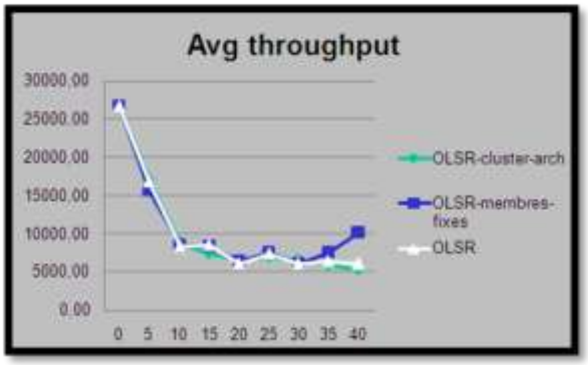

Figure 2- average throughput in term of node's velocity

In Figure 2, we note that in general, the flow remains the same for the three architectures

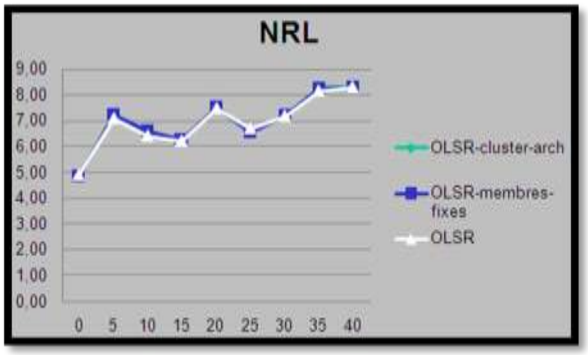

Figure 3- Normalized Routing Load in term of node's velocity

Figure 3 also shows that the NRL is unaffected and it remains the same for the different architectures, both in term of velocity or in term of the number of nodes

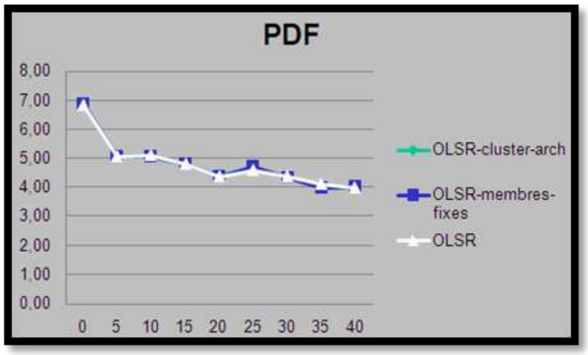

Figure 4- Packet Delivery Fraction in term of node's velocity

In figure 4 also we show that the PDF parameter is unaffected and it remains the same for the different architectures, both in term of velocity or in term of number of nodes

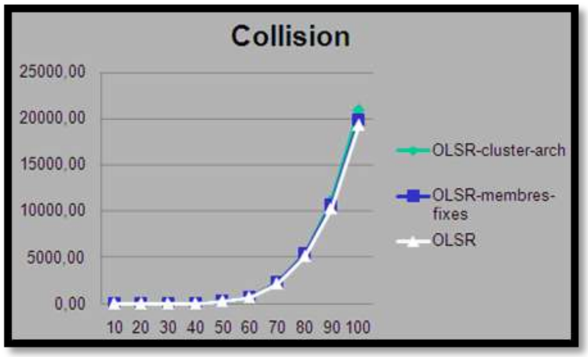

Figure 5- Collision based on the number of nodes

Similarly, in Figure 5 we note that collisions in the network remain the same even if the PKI architecture changes.

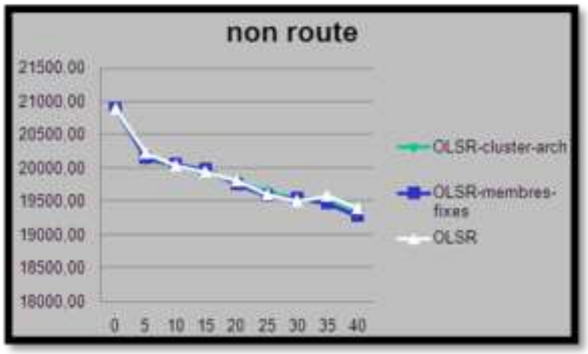

Figure 6- average non-routes in term of node's velocity

Like the other parameters, figure 6 shows that the number of non-routes is always the same.

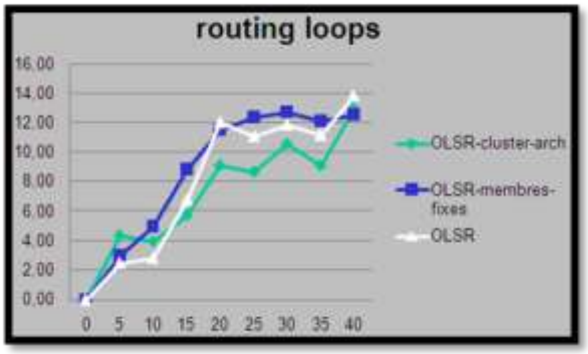

Figure 7- average routing loops in term of node's velocity

In Figure 7 we note a slight decrease in the number of routing loops in the network architecture for clustering

Generally, we can conclude that the operation of key management does not have bad effects on network performance. In the next section, we'll present the measurements of the second phase of this stage of our work. These measures represent some performance indices that we used to assess the reliability of our key management solution.

We measured the influence of the threshold parameters (k and n) in order to observe the behavior of the performance of the PKI. We recall that k represents the threshold contributions to validate a certificate, and n is the maximum of members of the PKI council.

The second phase results are as follows.

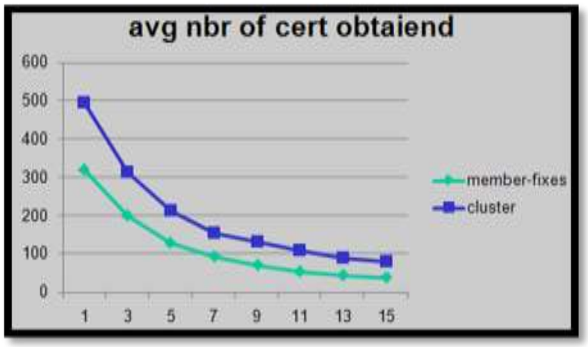

Figure 8- number of certificates obtained in term of k

Figure 8 shows the effect of parameter k on the number of certificates issued during the simulation for both fixed membership architecture, and clusterized architecture. For both architectures, we notice that as the threshold k increases, the number of licenses issued decrease. But the clusters based solution delivers more certificates than the fixed-member architecture.

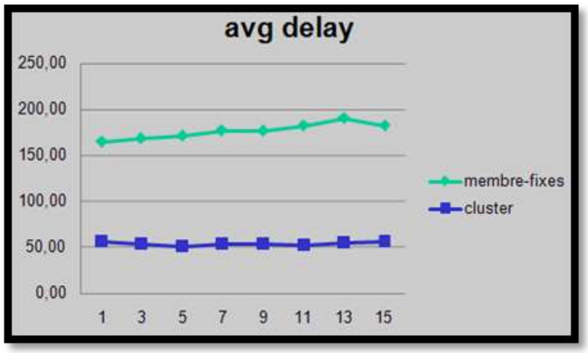

Figure 9- certificates delivery delay in term of k

Figure 9 shows the average delay to deliver a certificate. We note that the parameter k has no influence on the delivery of a certificate. However, the delay of the clustered architecture is 3.5 times lower than that of the fixed-member architecture.

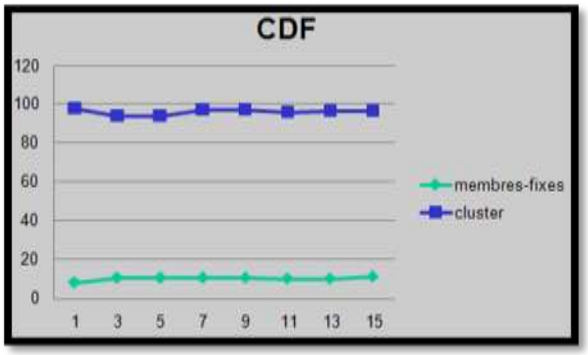

Figure 10- Certificates delivery fraction in term of k.

Figure 10 shows the certificates delivery fraction which represents the percentage of issued certificates by emitted certificates. We note that the parameter k has no influence on the CDF, and the cluster based architecture gives the better result than the fixed-member one.

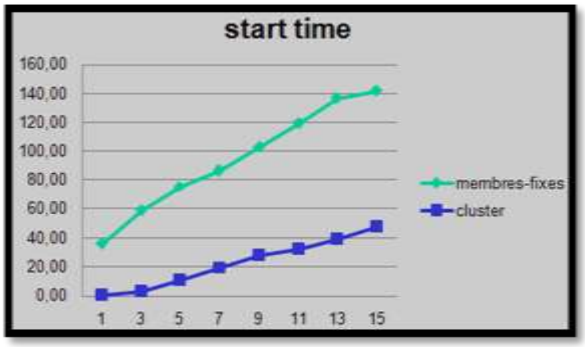

Figure 11- response time in term of k.

In figure 11 we show the PKI response time in term of threshold k. Response time (or start time as shown in the figure) represents the time elapsed between the beginning of the simulation and the delivery of the first certificate. For both architectures, the response time grows when the number k grows. But response time for cluster-based architecture is less than the fixed-member one.

In the next section, we'll present the results of the effect provoked by the parameter n (which represents the number of council members) on the performance of the PKI.

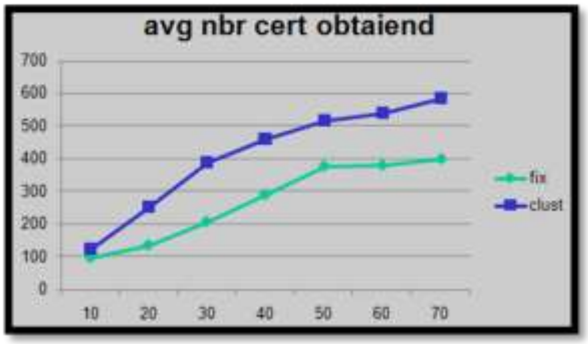

Figure 12- nbr of certificates obtained in term of n.

Figure 12 shows the effect of parameter n on the number of certificates issued during the simulation. Gradually as the number of members of council increases, the number of certificates issued increases.

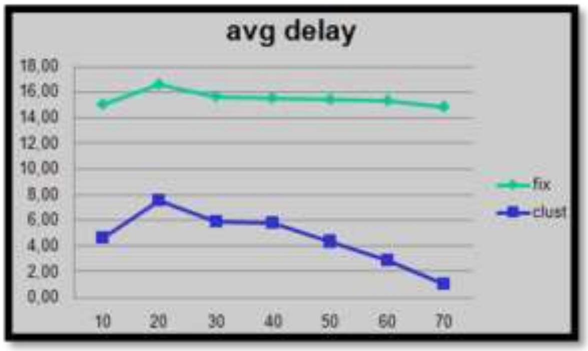

Figure 13- certificates delivery delay in term of n

Figure 13 shows the average delay to deliver a certificate. We note that the parameter n has less influence on the delivery of a certificate. However, the delay of the clustered architecture is around 2.6 times lower than that of the fixed-member architecture.

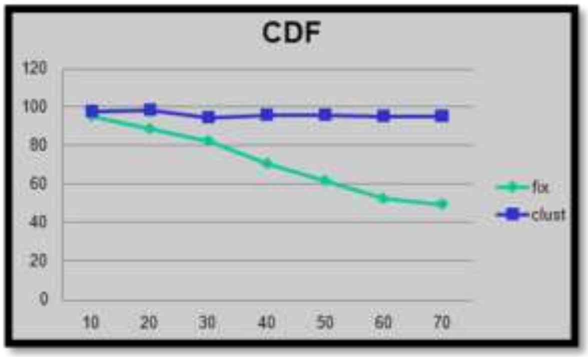

Figure 14- Certificates delivery fraction in term of n.

As shown in figure 14, in term of CDF, the parameter n has no effect on the cluster based architecture when it decreases the CDF in the case of the fixed-member architecture.

VII. Robustness Of The PKI

To measure the robustness of our PKI, we have tested to possible attacks. So we simulate two different types of attacks. The first type consists of a black-hole attack [21], in which the attacker node absorbs the traffic and do not let it pass from one network area to another. This way the network will be decomposed into inaccessible areas. The second type of attack consists of a grey-hole attack [21], in which the attacker modifies the control traffic by false data before to rebroadcast it from one side of the network to another. In this way, the network topology information becomes wrong, causing a loss of data packets.

We perform three simulations. The first simulation concerns an OLSR network without PKI in which attackers make an attack of black-hole that we have appointed "attack 1 OLSR" on the graphs that follow. The second simulation concern an OLSR network without a PKI in which attackers make an attack of gray-hole that we have appointed "attack 2 OLSR" on graphs. Third model of simulation concerns an OLSR network with PKI in which the PKI structure reacts with any type of attack by ignoring the malicious packets.

We measure some performance parameters of the network to observe the behavior of the proposed architecture against the attacks that we simulate. In what follows we present the results we have achieved.

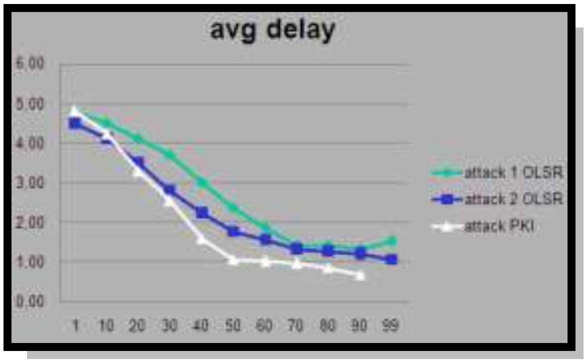

Figure 15- end to end delay in term of percentage of bad nodes.

Figure 15 shows the average end to end delay in the network. We note that the delay is improved in the case of a network with PKI, so it remains high in the absence of PKI.

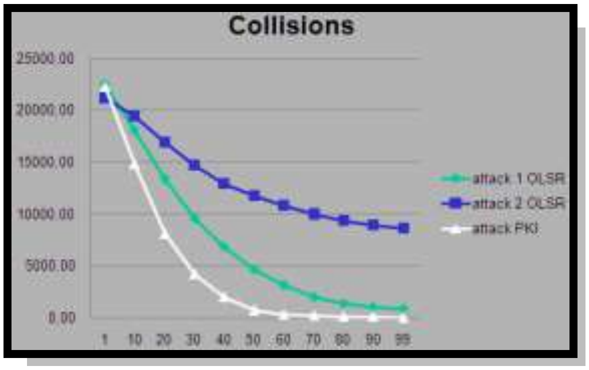

Figure 16 - number of collisions in term of percentage of bad nodes.

In figure 16 we show the number of collisions in the network. We note that the architecture with PKI generates a less amount of collisions than the case without PKI.

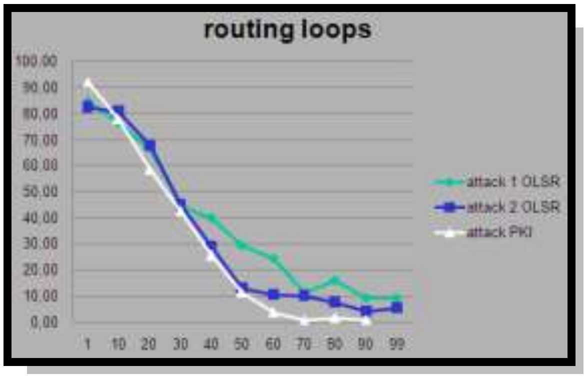

Figure 17- number of routing loops in term of percentage of bad nodes.

In figure 17, we show the number of routing loops in the network. We note that in an OLSR network with our PKI, routing loops are lower than in a network directly exposed to this kind of attacks.

Generally, we conclude that the PKI architecture that we proposed has a good robustness to different types of attacks that we simulate, and allows to optimize network resources in term of delay and bandwidth by eliminating unwanted traffic.

VIII. Conclusion And Perspectives

In a PKI (Public Key Infrastructure), the most important component is the CA (Certificate Authority), which is the trusted entity in the system that vouches for the validity of digital certificates. The success of PKI depends on the availability of the CA to the nodes in the network since a node must correspond with the CA to get a certificate, check the status of another node's certificate, acquire another node's digital certificate, and so on.

However, connectivity, which was assumed to be good in previous wired PKI solutions, is no longer stable in ad hoc networks. Unfortunately, maintaining connectivity is one of the main challenges in ad hoc networks, in view of the fact that the inherent infrastructurelessness of ad hoc networks makes it hard to guarantee any kind of connectivity. In this work, we presented a solution of cryptographic key management for ad hoc networks, based on a clustered architecture. In fact, the difficulty of finding a stable and permanent entity to ensure the function of CA requires distributing this role on all nodes in the network. But the problem of availability of all nodes simultaneously may cause the unavailability of AC services. For this, inspired by the threshold cryptography we have proposed creating a council of PKI that is composed of a subset among all nodes of the network.

Now to choose the council members we have proposed two solutions. The first solution is to designate a set of nodes that will make the council of PKI. These members are chosen randomly, and they remain the same as the network exists. That is what we mentioned in our article by fixed-members architecture.

The second solution is to organize the network in the form of clusters and each cluster will be represented by its cluster-head. All cluster heads of the network form the council of PKI. That is what we mean by Cluster-based Architecture.

We compared these two architectures and we figured out a set of measures that show that the clustered architecture provides a better result and is well suited to the dynamic environment of mobile ad hoc networks.

As perspective to this work, we plan to develop some aspects, focusing on the choice of the threshold parameter values (or k), and the council members number (or n).

References

[1]. A. Hajami, K. Oudidi, M. Elkoutbl, 'A Distributed Key Management Scheme based on Multi hop Clustering Algorithm for MANETs', IJCSNS International Journal of Computer Science and Network Security , Vol. 10 No. 2 pp. 39-49

[2]. A. Hajami, K. Oudidi, M. Elkoutbl, 'An enhanced algorithm for MANET clustering based on Multi hops and Network Density' NOTERE 2010, 31 Mai, 2 Juin 2010 Tozeur, Tunisie

[3]. T. CLAUSEN ET P. JACQUET. *Optimized Link State Routing Protocol (*OLSR*)*.http://www.ietf.org/rfc/rfc3626.txt, 2003, RFC 3626

[4]. Del Valle Torres Gerardo, Gómez Cárdenas Roberto. Proceedings of ISADS 2005. 2005 International Symposium on Autonomous Decentralized Systems (IEEE Cat. No. 05EX1007)

[5]. Kârpijoki Vesa, Security in Ad Hoc Networks., Telecommunications Software and Multimedia Laboratory 2002

[6]. L. Zhou and Z. J. Haas, .Securing Ad Hoc Networks., IEEE Networks, Volume 13, Issue 6 1999

[7]. H. Luo and S. Lu, .Ubiquitous and Robust Authentication Services for Ad Hoc Wireless Networks., Technical Report 200030, UCLA Computer Science Department 2000

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

[9]. S. Garfinkel, PGP: Pretty Good Privacy, O.Reilly & Associates 1995,ISBN 1-56592-098-8

[10]. S. Basagni, K. Herrin, E. Rosti and Danilo Bruschi, .Secure Pebblenets., ACM 2001

[11]. D. Balfanz, D. K. Smetters, P. Stewart and H. Chi Wong, .Talking To Strangers: Authentication in Ad-Hoc Wireless Networks., Internet Society, Conference Proceeding of NDSS Conference 2002

[12]. Kärpijoki Vesa, .Security in Ad Hoc Networks., Telecommunications Software and Multimedia Laboratory 2002

[13]. A.Shamir, "How to Share a secret", Communication of the ACM, Vol. 22, pp. 612-613, November 1979

[14]. C. Asmuth anf J. Bloom, "A Modular Approach to Key Safeguarding", IEEE Trans. On Information Theory, IT-29, pp. 208-211, 1983

[15]. E. F Brickell, "Some Ideal Secret Sharing Schemes", Journal of Combinatorial Mathematics and Combinatorial Computing, No. 6, pp. 105-113, 1989.

[16]. E. D. Karnin, J. W. Greene, and M. E. Hellman, "On Secret Sharing Systems", IEEE Trans. On Information Theory, IT-29, pp. 35- 41, 1983

[17]. T. P. Pederson, "Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing ", Lecture Notes in Computer Science, pp. 129-140, 1992

[18]. Y. Desmedt and S. Jajodia, "Redistribution Secret Shares to New Access Structures and its Applications", Technical Repport ISSE TR-97-01, George Mason University, Fairfax, VA, July, 1997.

[19]. Vivek Shah, "Parallel Cluster Formation for Secured Communication in Wireless Ad hoc Networks", work submitted as part of requirement for the degree of Master of Science in Computer Science. University of Cincinnati. 2004

[20]. A. UC Berkeley and USC ISI, "The network simulator NS-2", Part of the VINT project, Available from www.isi.edu/nsnam/ns, 1998.

[21]. Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In Proceedings of the 8th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '02), September 2002.

[22]. A. Patwardhan, J. Parker, A. Joshi, M. Iorga, and T. Karygiannis. Secure Routing and Intrusion Detection in Ad Hoc Networks. Third IEEE International Conference on Pervasive Computing and Communications, Kauaii Island, Hawaii, March, pages 8–12, 2005.