Toward a Peer-to-Peer PKI for Mobile Ad-Hoc Networks

Hella Kaffel-Ben Ayed, A. Belkhiri

Abstract—**Deploying PKIs in ad hoc networks opens up various issues related to the intrinsic characteristics of these networks. In the literature, many proposals for PKI over ad hoc networks are based on the distribution of the certification authority via a threshold secret sharing scheme. However, these proposals are mostly suitable for managed ad hoc networks. In this paper, we propose a self organized peer-to-peer CA. This CA is generic and can suit various contexts of spontaneous ad hoc networks. It does not rely on any central or external entity. The CA's services are carried out by the different participating CA members determined. The proposed protocol has two phases: the bootstrapping phase and the operating phase. The bootstrapping constitutes our main contribution compared to related works. We evaluate by simulations our proposal and show that its performances are acceptable while considering various scenarios for ad hoc networks.**

Index Terms—**PKI, certification authority, ad hoc networks, Peer-to-Peer.**

I. Introduction

MANET (*Mobile Ad Hoc Networks*) are endowed with many virtues which make them very interesting in both military and civilian fields. Thanks to their intrinsic characteristics (no pre-deployed infrastructure, a dynamic topology, an open transmission medium), they are highly effective in numerous situations such as emergency and rescue. At the same time and because of these features, MANETs are prone to a wide range of attacks which may range from the simple eavesdropping to the breakup of some vital functions of the network (such as routing). Cryptographic techniques are often seen as the most effective tool providing networks with security services [15],[20]. The use of cryptographic schemes (either for encryption or for signing) relies greatly on a secure and effective key management [32]. Most recent researches on MANETs security address the issue of setting up a secure and robust PKI (Public Key Infrastructure). Unfortunately, the aforementioned features of mobile ad hoc networks added to the lack of a centralized administration authority, the error-prone transmission medium and the vulnerability of mobile nodes to physical attacks, has rendered the task of setting up such framework very hard.

The last decade has seen an effervescence in MANET's security research domain. However, we agree with the authors of [28, 21] that this research domain still immature. The proposed schemes of PKI over MANETs are not suited to fully self organized ad hoc networks. This can be easily explained by the fact that most of these proposals are based on the existence of an offline CA (Certification Authority), commonly known as *Trusted Dealer*, which provides some nodes with keying materials before the network is set up [40,41,42,43]. We propose in this paper a spontaneous self organized peer-to-peer CA. This CA can suit the context of spontaneous mobile ad hoc networks.

The remaining of this paper is organized as follows: In section II we give a brief presentation of the most effective proposals for deploying a PKI for MANETs. In section III, we present and discuss our solution. Section IV portrays the evaluation of our proposal through simulations. Finally, in section V we conclude this paper and outline some future works.

II. Related works

The deployment of a PKI in mobile ad hoc networks has been considered as a challenging task. Researchers have noticed that, contrary to classical wired and static networks, it is impossible for a single node in a MANET to play the role of a CA because of the aforementioned security weakness of mobile nodes. Moreover, researchers have proved that even if it were possible, nodes could be unable to get the certification services because of the connectivity transience. To solve this issue, various solutions have been proposed aiming to reach a tradeoff between security and availability of the certification services [1],[2],[4],[6]-[10],[16],[17], [20],[21],[23]-[26],[28],[29],[33],[37]-[45],[47]. In this section, we review some of these proposals. We classify them into four categories according to their architecture.

A. The Partially distributed PKIs

A first scheme based on a partially distributed certification authority was proposed in [43]. The CA's functionalities are

Manuscript received March 31, 2011

H Kaffel-Ben Ayed is Assistant Professor at The Faculty of Sciences of Tunis, University of Tunis El Manar and Senior researcher at CRISTAL Lab., National School of Computer Science, University of Manouba. Tunisia. Hella.kaffel@planet.tn

A.Belkhiri is a Master student at the Faculty of Science of Tunis, University of Tunis El Manar, adel.belkhiri@gmail.com

distributed over a random set of nodes by using threshold cryptography. This paradigm uses a *(t, n)* secret sharing scheme [35] to distribute cryptographic operations over *n* different players. Such scheme ensures that the required operation will be infeasible unless there is the participation of at least *t* players. To get its certificate signed by the distributed CA, a client node (requesting for a certificate) must send to server nodes his public key with its credentials contained into a certification request. Each server node receiving such request generates a partial signature using its share of the CA's private key and submits it to a special node (called *Combiner*) that combines the *t* partial signatures into a valid one. The validity of the generated signature can easily be verified using the public key of the distributed CA.

In order to enhance the performances of this scheme, many proposals have been published later by Yi in [40, 42] and by Wu in [39]. Yi's proposals suggest the distribution of the functionalities of the CA over the most powerful, secure and trustworthy nodes in the MANET. Yi called these nodes *MOCA* (*MObile Certificate Authority*). To contact this distributed CA, many strategies are proposed. They are based on the idea that MOCA certification protocol can share routing information with the available routing protocol. Hence, a client node may contact MOCA servers by using, among many others strategies, either the shortest or the freshest paths in its routing cache.

Since a client node has to ask many sever nodes for the certification service at the same time, an availability issue may be raised. Wu proposes in [39] a scheme called *SEKM* (*Secure and Efficient Key Management*) which enhances the availability of the distributed CA. Server nodes remain connected by sending periodic messages between each others. To get its certificate signed by the distributed CA, a client node sends a certification request to at least one server node. Thus, the availability of the whole scheme is enhanced since each server node has a view of the whole distributed CA. However, all these proposals share the following limitations:

1. Since the nodes which are members of the distributed CA are chosen by the trusted dealer, these schemes are only suitable for managed ad hoc networks.

2. It is the responsibility of server nodes to store all issued certificates in the network. In case of limitation of nodes' memory, this issue may lead to a memory space problem.

3. These schemes are not able to scale with the network's size since the parameters *t* and *n* of the threshold secret sharing scheme on which they are based are fixed in advance.

B. Fully distributed PKIs

In references [23]-[25], a new approach for distributing CA functionalities is proposed. It enhances the availability of Zhou's proposal [43]. But unlike Zhou's approach, these solutions use a *(t, n)* threshold secret sharing scheme to distribute the CA's services over all the nodes in the MANET. Thus, being based on a localized trust model, the certification services in the fully distributed CA approach can be performed by any *t* one-hop neighboring nodes. It is important to mention here that the offered certification services do not provide the initial certificates issuing. In addition to initializing the first *t* nodes by sharing the CA's private key among them, initial certificates issuing task is also accomplished by the trusted dealer. To renew its certificate, a node must broadcast a request to a coalition formed by at least *t* one-hop nodes in its neighborhood. Each requested node, based on the client node's behavior which is monitored by a local intrusion detection system, uses its private key share to make a partial signature over the certificate. This proposal fits better the ad hoc network's constraints, since the burden of ensuring the security of the network is fairly shared by all the nodes. However, it inherits the weaknesses of schemes discussed in subsection A, in addition to the following limitations:

1. Authors assume in their model that every node can have at least *k* one-hop neighboring nodes. This assumption is often unrealistic and a node may have less than *k* one-hop neighbors.

2. Since initial certificates issuing is not ensured by the proposal, the knowledge in advance of all participating nodes identities seems necessary. This issue makes the proposal unsuitable for spontaneous unplanned ad hoc networks.

3. The proposal does not consider how to adjust the parameter *t*: a too high would affect the service availability, and a too low would affect the security of the system. Moreover, it is not shown how this parameter (*t*) can be adapted to the size of the network.

C. Certificate chaining-based PKIs

Starting from their point of view that security in MANET must not rely on any TTP, even throughout bootstrapping phase, Capkun and Hubeau [20], [37] have shown that public key management can be done in a fully self-organized fashion. In their proposals, inspired of the PGP authentication system [46], digital certificates are created, signed, issued and stored by nodes themselves. Based on its belief that a public key *PKv* belongs to a specific node *v*, a node *u* can issue a certificate that states such "ownership". In [20], [37], the authors present their scheme as an oriented graph model *G(V,E)* where *V* (*Vertices*) correspond to public keys and *E* (*Edges*) are associated to the issued certificates. The certificates are selected according to *Shortcut Hunter* or *Star Shortcut Hunter* algorithms. Each node stores in its local repository a small number of certificates that have been issued. This repository constitutes the node's local view of the whole graph G. To authenticate their public keys, communicating nodes proceed as follows: they first merge their local certificate repositories (named web-of-trust), then they search in the merged repositories for a certificate chains between them. Schemes based upon certificate-chaining approach share the following limitations:

1. The authentication of public keys cannot be guaranteed. Indeed, a certificate chain between two nodes may not be found (the graph representing the trust relationships between nodes may not be dense enough or not connected.

2. A long time period is required for until nodes can establish a web-of-trust between each other.

3. Since these schemes are not based on any kind of TTP, expected results could not be accurate. Nodes, in such schemes, act like a standalone CA and therefore the validity of any certificate-chain will depend on the honesty of all nodes involved in its formation.

D. Cluster-based PKIs

Clustering has been often used to enhance the availability of the CA services and to minimize the use of the network's bandwidth. Then, various schemes use clustering techniques to setup PKIs over ad hoc networks [1], [2], [9], [10], [16], [17], [21], [28]. Because clustering techniques are used differently and for various purposes, each scheme will be briefly described apart.

Authors in [28] use clustering techniques to take advantage of the neighbors' monitoring capabilities and the short communication range within the same cluster.

Inspired of the "web-of-trust" approach, authors assume that nodes belonging to the same cluster are able to establish a direct trust relationship with each other by using behavior monitoring systems. The authors define the concept of *Introducing nodes.* These are outsider nodes with which a requesting node had yet trust relationships. They belong to the same cluster as the requesting node. Based on many "signed recommendations" issued by these outsider nodes, a requesting node may establish indirect trust relationships with nodes from other clusters. The problem here is that a given node may have to authenticate a node from a foreign cluster without an introducing node.

Authors in [1] propose to split the network into clusters. The set of clusterheads, which jointly constitutes the distributed CA, uses a proactive secret sharing scheme to distribute network's private key over them. To get its certificate signed by the distributed CA, a client node must collect some warranty certificates as credentials. Based on a *(t, n)* threshold signature scheme, a quorum of clusterheads collaboratively sign the client node's certificate after verifying the validity and the number of created warranty certificates. The main drawback of this scheme is that the certification services are assumed to be handled by the clusterheads without considering their trustworthiness or their ability to offer such services.

Authors in [17] proposed a self-organized key management in which clusterheads (called *CMNs* for *Certificate Management Node*s) collect and manage certificates issued by nodes in their one-hop neighborhood. This scheme has the advantage of optimizing certificates storage since a multiple *CMNs* share all the certificates in the network. Moreover, it reduces the traffic load since nodes (called *NN* for *Normal Nodes*) entrust the finding certificate-chains task to *CMNs* instead of proceeding by merging their repositories like in [20, 37]. However, this scheme suffers from two main limitations: 1)Similarly to Hubaux's scheme, the results given by this scheme cannot be guaranteed since *CMNs* may not find a certificate chain between two authenticating nodes. 2)Unlike Hubaux's scheme, heavy computation and storage load are carried only by the few *CMNs* nodes. This issue contradicts the concept of symmetric relationships between MANET's nodes.

Authors in [16] propose a composite key management by using various techniques: distributed CA, identity-based cryptography and certificate-chain authentication. Their scheme is mainly based on the availability of a trusted dealer (an offline CA) which is responsible for performing numerous vital functions such as creating clusters and selecting clusterheads, generating private/public key pair, creating a certificate for each clusterhead, registering new joining nodes, detecting topology changes, collecting reports from clusterheads, refreshing clusterheads key pairs, etc. . This scheme has the two following shortcomings: 1) It uses a clustering algorithm called *CGQR* (*Clusterhead Gateway Switch Routing*) which does not guarantee the trustworthiness of elected clusterheads. 2) The shared signature key is generated by a randomly selected clusterhead called *KM* (*Key Manager*). Assuming that this *KM* will be a trustworthy node does not seem a realistic assumption.

Authors in [2],[9],[10] use a secure clustering algorithm called *RECA* (*REputation based Clustering Algorithm*) [11] to elect trustworthy clusterheads and to distribute the CA's services among them. Each clusterhead has a twofold function: a centralized CA (for the members of its cluster) and a member of the distributed CA (for new elected clusterheads). Within the same cluster, nodes validate each other's certificates using the public key of their clusterhead. To validate the certificate of nodes from other clusters, a request must be sent to one's clusterhead which know the public keys of all clusterheads in the network and which can, therefore, verify its validity using the appropriate public key. The main drawback of this scheme is that if a malicious node succeeds to compromise just one clusterhead, it will be able to issue false certificates that would be recognized as valid by all the nodes in the network.

III. The proposal

MANETs have similarities with the peer-to-peer (P2P) networking model in several aspects: decentralization, equality and autonomy [6]. Hence, we propose a generic P2P certification authority without the intervention of a central or any offline entity. The CA's services are carried out by the different participating CA members. This CA can be set up anywhere at any time as soon as a spontaneous P2P network is created.

A. The requirements

For the design of an effective CA in a spontaneous peer-to-peer network, we define the following requirements:

1- *Non preestablished trusted dealer: P*reestablished trusted dealer solutions fit planned peer-to-peer networks where the identities of nodes are well known in advance. Furthermore, if the trusted dealer is usually well protected against external attacks, it is not the case against internal ones resulting in the disclosure of its private key.

2- *A trust anchor:* It is crucial for the credibility of the issued certificates that the CA is trusted by all the nodes in the network. To achieve this goal, CA nodes are chosen according to their honesty.

3- *The availability of the CA services:* this feature depends greatly on the participation of a sufficient number of nodes. The higher the number of participating nodes is, the more the certification services will be available.

To fulfill the identified requirements, we rely on clustering techniques as well as on threshold key generation schemes [13], [15], [31], [32]. In the following, we first present the network model. Then, we describe the features of the clustering protocol that is required by our scheme. Finally, we explain in more details our protocol.

B. The assumptions

Communication links between nodes are insecure: They are prone to a wide range of attacks that characterizes both wireless and peer-to-peer communications (like eavesdropping and *MITM* attacks for example). We assume that each node is endowed with a reputation system allowing it to assign for each one-hop neighbor a trust value. This system may be empowered by an Intrusion Detection System (IDS) that can be used for the detection of malicious nodes. Reputation system and IDS may cooperate by feeding each other with the relevant information in order to enhance their performances. Moreover, we assume that the network can be split, by using an appropriate clustering protocol, into many clusters*.* *W*e consider that nodes are mobile and can roam freely from one cluster to another.

C. *The clustering algorithm*

In our proposal we rely on clustering so as to distribute CA's services over the elected clusterheads. In our context, it is important for the efficiency of our scheme that the clustering algorithm takes into account the honesty of nodes while computing their weights (such as *Weighted Clustering Algorithms- WCA*) [5], [11], [22]. WCA requires that each node of the network is equipped with a *GPS* (*Global Positioning System*) to compute the positions of nodes while they are in move. This assumption may not be realistic in our context. The algorithms proposed in [22, 11], which are called respectively *SCA* (*Secured Clustering Algorithm*) and *RECA* (*REputation based Clustering Algorithm*), perfectly meet our needs. These algorithms take into account the following parameters for the election of clusterheads in a MANET:

- *The Max value:* maximum number of nodes which may be handled by a single clusterhead.
- *The Min value:* minimum number of nodes which may be handled by a single clusterhead.
- *The Max hop cluster:* maximum number of hops which may exist between a clusterhead and its cluster's members.
- *The Weight:* a node may be elected as a clusterhead according to its weight in the cluster. In order to compute node's weight the following parameters are considered: 1) *Trustworthiness* (computed according to the records of its one-hop neighbors reputation system) and 2) *Battery power* (the remaining lifespan of node's battery).
- *The Stability* of links between a given node and its neighbors. It is usually affected by the node's mobility and by the transmission range.

D. The protocol

As stated before, our approach eliminates any kind of trusted dealer. In our design, we use as aforementioned a clustering protocol to select a set of nodes which will form the online CA, and a distributed secret sharing protocol to share CA's private key among them. the process may be described according to two phases: a bootstrapping phase and an operating phase. By the end of the former one, the CA will be operational and able to offer certification services for ad hoc network nodes.

1) The bootstrapping Phase

This phase aims at setting up a distributed online CA within a mobile ad hoc network. It represents the peculiarity of our approach compared to other works. The bootstrapping phase begins when each node in the network establishes enough trust relationships with its one-hop neighbors. At that time, the network structure is flat. After the execution of the clustering algorithm, the network's structure becomes hierarchical and nodes will be grouped into many clusters managed by clusterheads as explained in subsection C.

We propose to distribute the functionalities of the CA over clusterheads which are considered trustworthy by their one-hop neighbors. These clusterheads will start providing the certification services for all nodes in the network. Clusterheads are equipped with a threshold signature scheme which enables them to share the capacity of signing certificates on behalf the CA: a set of t out of n clusterheads can cooperate to jointly sign a certificate for a client node.

In the literature, many threshold schemes have been proposed to share the signature function among a set of nodes [14], [18], [30], [36]. The secret key used in such schemes is generated and shared using secret sharing protocols. These schemes can be classified in two categories:

- Centralized: The secret key is generated and then divided into shares by one centralized trusted dealer [12], [32], [35]. Each share is sent to a server node by that dealer.
- Distributed: The secret key is generated and shared by server nodes themselves without the help of any outsider entity in a distributed manner. By the end of the protocol, commonly known as DKG (*Distributed Key Generation*) protocol, each server node will have a share of the secret key but none of them will have knowledge of the secret key itself.

As it was stated before, whenever the secret key is entirely owned by a single entity, the security of the whole system is jeopardized. Starting from this fact, we chose the distributed approach for the key generation. Several protocols, in literature have been proposed to generate the public/private key pair for threshold RSA based cryptosystems [3], [15] as well as for discrete logarithm based cryptosystems [13], [31], [32], [34]. Although most of DKG protocols assume the existence of private channels between each couple of server nodes, which means obviously that cryptographic materials are already deployed on such nodes, the one proposed in [13] does not. Besides having the advantage of being non-interactive, this *DKG* protocol uses only public channels. We have adopted for this latter protocol since it fits our reuirements. Each clusterhead has to execute this protocol to get its share of the distributed CA's private key SKca in addition to its public key PKca. Once a clusterhead has obtained a share of the CA's signing key, it cooperates with other clusterheads to jointly generate and sign a certificate authenticating the CA's public key *PKca*. The most important information carried in the certificate are the validity period, the CA's public key *PKca* and the CA's signature.

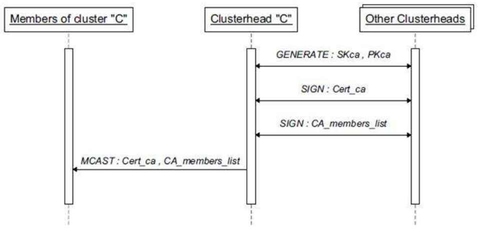

Fig. 1. The sequence diagram of the Bootstrapping phase.

The process of signing the CA's certificate is the same as that triggered in response to receiving a certificate signing request. To ensure that each clusterhead has already computed its own share, a message of synchronization between clusterheads is used. The CA's certificate is distributed by each clusterhead to the members of his cluster. In the same way, clusterheads cooperate together to sign with the CA's private key *SKca* a list containing the identities of the CA members. This list is sent with the auto-signed certificate of the distributed CA by all clusterheads to their respective clusters members (see Fig. 1). Besides, each clusterhead carries out the certification of its own key pair (*Pk/Sk)* by requesting the cooperation of other clusterheads. Then, clusterheads exchange their certificates and send them to nodes in their own clusters. In order to maintain a good level of security, it is important to refresh the shares which are held by clusterheads each time the group forming the distributed CA's changes (a clusterhead leaves it or a new one joins it) [19]. Accordingly, the CA certificate has to be re-generated and re-signed by CA's members.

2) *The Operating phase*

To prevent a malicious node, in case it has been elected as a clusterhead (after launching a *Sybil attack* for example), from signing certificates on behalf of the CA, it is wise to make that task only possible for a set of clusterheads. The signature function is therefore shared among clusterheads according to a *(t , n)* threshold signature scheme. Such scheme allows each CA's member to generate a partial signature in response to a certification request. We have chosen the threshold signature scheme described in [30]. The latter is a distributed variant of the DSA (*Digital Signature Algorithm*).

- *The Certificate issuing service:* To get its certificate signed by the distributed CA, a client node must first target the CA's members that are able to serve it by sending a service request to a quorum of *t+Δ* clusterheads (cf. Fig.2). The *Δ* value represents an estimation of the number of clusterheads that may be unable to serve, at that time, for one reason or for another (lack of resources, being under DoS attack, etc.). The client node must, after that, pick up clusterheads identities from which it has received a "*service engagement*". If the number of responses outnumber the threshold *t*, this latter node sends them a certification request. The following information have to be conveyed by the certification request: client's public key, credentials (*Cert_cli* field), identities of clusterheads (*CH_*ids) that have accepted to certify it and other information related to its identity. By applying the threshold signature scheme, each clusterhead will be able to compute and to send a partial signature *PS* to the requesting node. Once partial signatures are received, the client node checks their validity and combine *t* ones out of them into a complete signature. The validity of the final signature may be checked by using the CA's public key *PKca* (Fig.2).

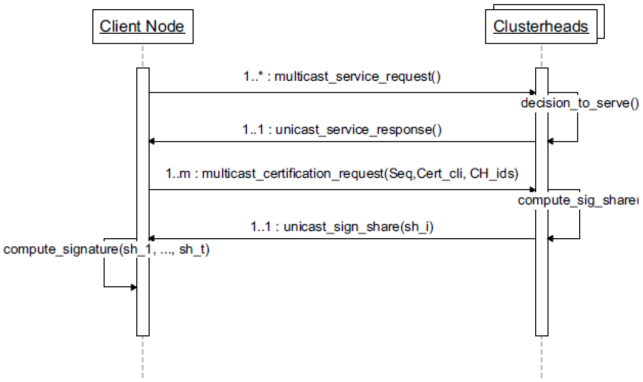
Client Node
Clusterheads
1..* : multicast_service_request()
decision_to_serve()
1..1 : unicast_service_response()
1..m : multicast_certification_request(Seq,Cert_cli, CH_ids)
compute_sig_share(
1..1 : unicast_sign_share(sh_i)
compute_signature(sh_1, ..., sh_t)

Fig. 2. The Certificate issuing process.

- *Certificate publishing service:* In order to publish the certificates issued by the distributed CA, each node has to send its certificate to its clusterhead. Periodically, clusterheads exchange the identities of the members of their clusters as well as the levels of their honesty. If a given node *Ni* looks for the certificate of another node *Nj*, it sends a request to its clusterhead. This latter determines the clusterhead of *Nj* and forwards to it the request. Once the clusterhead associated to

Ni receives the certificate of *Nj* it forwards this certificate to *Ni* with the corresponding trust level.

- *Certificate revocation service: A* node's certificate has to be revoked if the private key of the node has been compromised or if its trust level becomes below a given threshold. Each clusterhead maintains, in a local repository, the list of all revoked certificates. This list is periodically exchanged between clusterheads. To check the status of a given certificate in an online way, a node has to send a request to its clusterhead.

IV. SIMULATIONS AND RESULTS

For the evaluation of our proposal we consider mobile ad hoc networks as a case study for the deployment of the proposed P2P certification authority. However, our proposal can be used to support any other type of P2P spontaneous networks. The performance of our scheme is evaluated by simulation with ns2 under a UNIX platform. We have used a laptop an Intel Centrino dual core 2.4 GHz and a RAM and a 4 GB memory. Since certification services costs were largely evaluated in the literature, we focus here on the bootstrapping phase. The simulation presented here cover only the communication aspects of the CA's signing key sharing phase. For this end, we use of two metrics:

- the *Average delay:* the time required for each clusterhead to get its share of the CA's signing key.
- The *Overhead:* The number of messages exchanged per second throughout the aforementioned phase.

We have varied the following parameters: the number of clusterheads, the transmission range, the speed of mobile nodes and the area of ad hoc network.

Each point on curves presented in this section is the average of 10 simulation runs. We estimated a 95% confidence interval of each performance measure. Error bars are not drawn for the clarity of figures. Since the implemented protocol requires computation over huge numbers, we have used of the *gmp library* (*GNU Multiple Precision arithmetic library*). The scenarios were generated using the parameters which are listed in Table 1:

TABLE 1

PARAMETERS OF THE SIMULATIONS

Parameters	values
Network simulation area	200×200→1000×1000 m^2
Number of nodes	100
Mobility model	Random Waypoint
Routing protocol	AODV
Transmission range	70 → 200 m
Max speed	1 → 10 m/s
Pause time	10 s
Simulation time	600 s

A. *Impact of varying the number of clusters*

We evaluate here the impact of varying the number of clusters on the performance of our scheme during the bootstrapping. A comparison between the costs of bootstrapping with and without trusted dealer is made. We have considered scenarios similar to those adopted in [11]. In Fig. 3 and in Fig. 4, we consider that nodes move randomly in an area of 1000×1000m^2 at speed of 5m/s and have a transmission range of 200m. Nodes in the network are administrated by a set of clusterheads whose number varies from 2 to 16. Fig. 3 depicts the variation of the average delay that each clusterhead has to wait until it can compute its share of the CA private key. Fig. 4 portrays the induced overhead, depending on the total number of clusterheads. As it is shown by the shape of the curve, the higher the number of clusterheads becomes, the higher will be the average delay and the overhead. Nevertheless, the impact of clusterheads' number on the average delay of bootstrapping using a trusted dealer is insignificant. This seems logical since in a DKG bootstrapping the communication between server nodes is accomplished in an n-to-n fashion whereas in a dealer-based bootstrapping a 1-to-n communication is initiated.

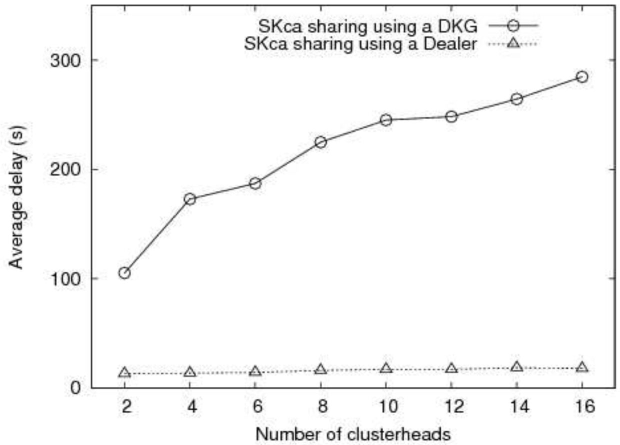

Fig. 3. The average delay vs. number of clusterheads during the SKca sharing phase

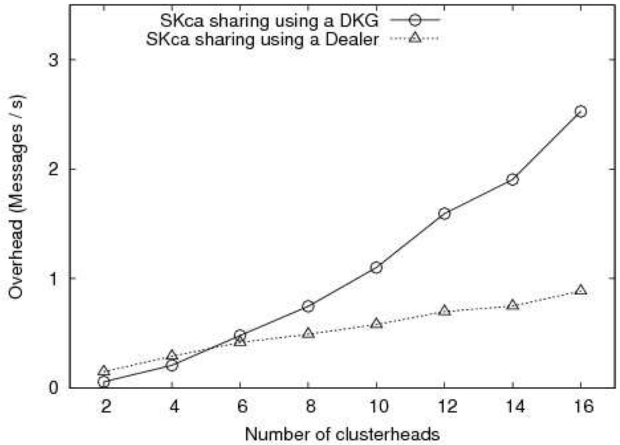

Fig. 4. The overhead vs. number of clusterheads during the SKca sharing phase

B. Impact of varying transmission range

We show here the adequacy of our approach to nodes with different transmission capabilities. Thus, by varying the transmission range of nodes, we depict how our metrics vary depending on the connectivity and batteries lifespan. In Fig. 5 and Fig. 6, we consider 100 nodes grouped into 16 clusters. These nodes are randomly moving in an area of 500 × 500m^2 at a speed of 3m/s. The shapes of curves in Fig. 5 and Fig. 6 show that nodes' transmission range affects slightly the average delay and the overhead of the bootstrapping phase. Fig. 5 shows that the average delay for both DKG approach and dealer-based approach are slightly decreasing until a certain transmission range value (170m for the former and 130m for the latter) and then they increase slowly. This phenomenon may be explained by the routing protocol's behavior which is affected by the number of collisions and the frequency of one-hop links establishment. Thus, since the impact of transmission range on the performance of our scheme is not considerable, we can state that our scheme is suitable for different configurations of ad hoc networks either those giving priority to nodes' connectivity or those giving priority to economizing batteries' power.

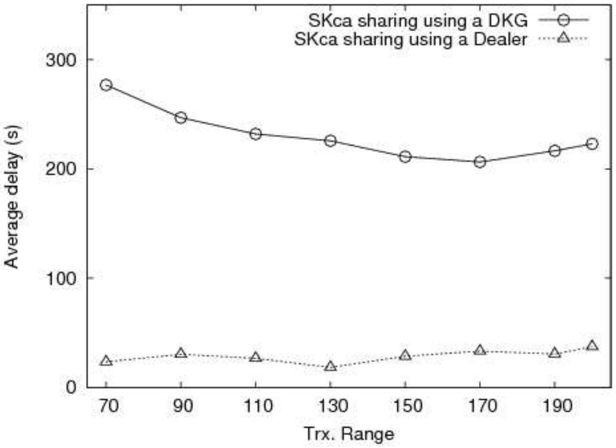

Fig. 5. Average delay vs. the nodes' transmission range for the SKca sharing phase.

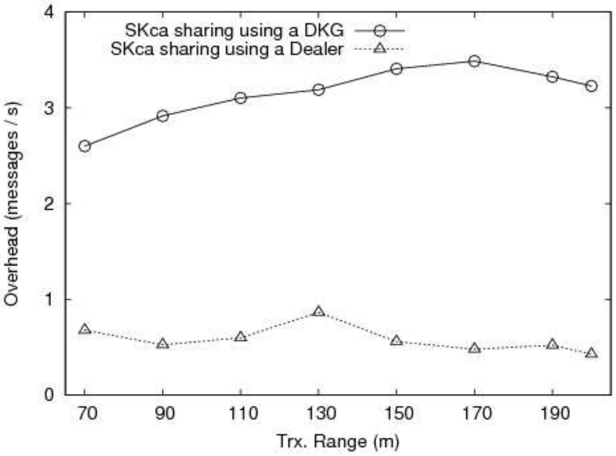

Fig. 6. The overhead vs. the nodes' transmission range for the SKca sharing phase.

C. Impact of varying nodes velocity

We intend through this scenario to evaluate the impact of nodes' speed, within the ad hoc network, on the metrics we have chosen. We consider 100 mobile nodes which are moving in an area surface of 1000 × 1000 m2 and having each a transmission range of 200 m. Nodes are able to move at many speed levels ranging from 1m/s to 10m/s. We consider 4, 8, 12 and 16 clusters. Fig. 7 and Fig. 8 show that the nodes' speed has not a considerable impact on the average delay and on the overhead of the bootstrapping phase. These results reflect the impact of nodes' speed on the performances of the used routing protocol. So we can affirm that our scheme is suitable for different contexts of mobility.

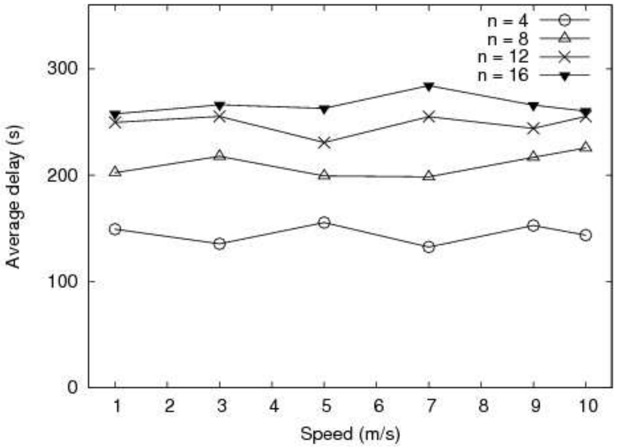

Fig. 7. The average delay vs. the nodes' speed for SKca sharing phase.

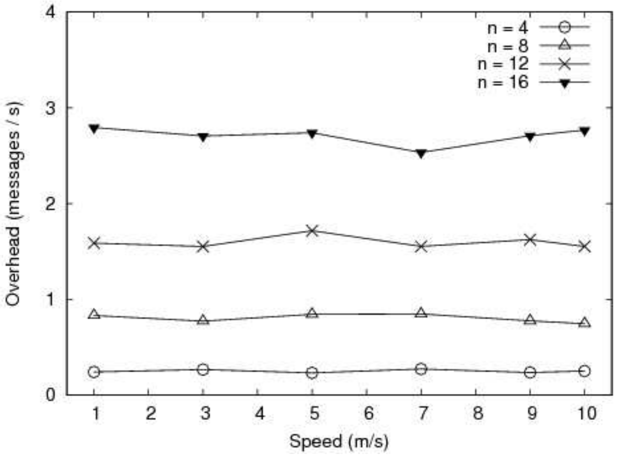

Fig. 8. The overhead vs. the nodes' speed for SKca sharing phase.

D. Impact of varying the simulation area

These simulations aim to evaluate the impact of the variation of the density of the network on the performance of the bootstrapping phase. Fig. 9 and Fig. 10 present the variation of the average delay and the overhead depending on the ad hoc network's area. We consider here that nodes move randomly at a speed of 5m/s in various areas ranging from 200×200 m^2 to 1000×1000 m^2. We have fixed the transmission range to 200 m for each mobile node. Fig. 9 shows that the average delay increases for a certain area (600 × 600 m2). Respectively, in Fig. 10 the overhead decreases beyond this area since the chosen criteria are correlated. Indeed, in large areas, nodes can be frequently out-of-range of each other. This affects the routing function in the network and results in the increase of the bootstrapping duration.

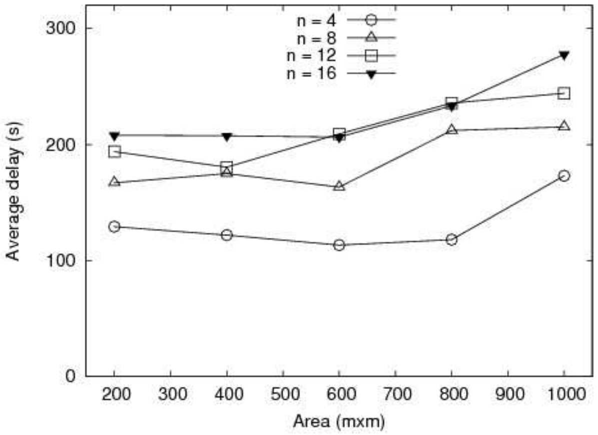

Fig. 9. The average delay vs. the simulation area for SKca sharing phase.

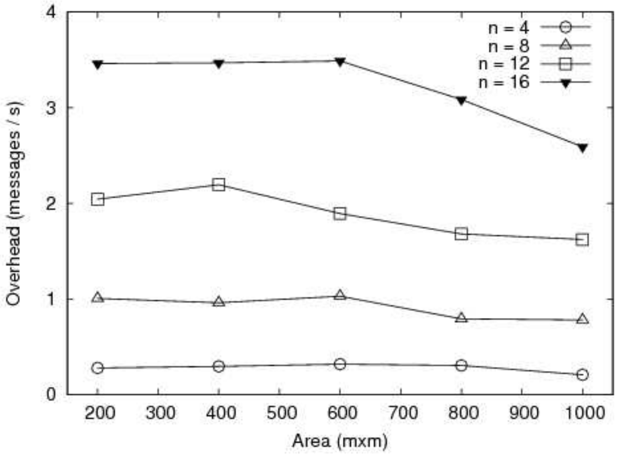

Fig. 10. The overhead vs. the simulation area for SKca sharing phase.

V. CONCLUSION

Our main contribution in this paper consists in proposing the design of a generic P2P CA over a mobile ad hoc network without relying on a trusted dealer. We focused on the bootstrapping phase of our protocol. Simulations show that the proposed solution is suitable for different configurations of ad hoc networks either those giving priority to nodes' connectivity or those giving priority to economizing batteries' power and for different contexts of mobility. The simulations show that the average delay increases in large areas. Furthermore, the time required for the bootstrapping of our scheme is larger than in dealer-based schemes. However, since the bootstrapping phase is occurring just once in the whole network's lifespan, this latency time can be tolerated especially because of the main advantage of our proposal: providing certification services in situations where the deployment of the ad hoc network cannot be planned in advance. Ongoing work focuses on evaluating this approach over various types of spontaneous P2P networks.

ACKNOWLEDGMENT

We are grateful to Mrs SOUISSI Donia for the care with which she reviewed this original manuscript.

REFERENCES

[1] M. Bechler, H.-J. Hof, D. Kraft, F. P.hlke, and L. Wolf, "A cluster-based security architecture for ad hoc networks," In *Proc. INFOCOM, 2004,* volume 4, pp. 2393–2403.

[2] L. Benazzouz, M. E. Elhdhili, and F. Kamoun, "Towards an efficient reputation based hybrid key management architecture for ad hoc networks," *Security and Communication Networks*, 2010, 3(2-3):261–277.

[3] D. Boneh and M. Franklin, "Efficient generation of shared rsa keys," In *Advances in Cryptology–CRYPTO 97* Springer-Verlag, 1997, pp. 425–439.

[4] P. Caballero-Gil and C. Hern.ndez-Goya, "Self-organized authentication in mobile ad-hoc networks," *Journal of Communications and Networks*, 2010, vol.11, 509–517.

[5] M. Chatterjee, S. K. Das, and D. Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks," *Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks)*, 2001, vol.5, pp.193–204.

[6] H. Deng, A. Mukherjee, and D. P. Agrawal, "Threshold and identity-based key management and authentication for wireless ad hoc networks," In Proc. *International Conference on Information Technology: Coding and Computing(ITCC 2004), 2004.* Pp. 107–115.

[7] A. W. Dent and G. Price, "Certificate management using distributed trusted third parties," In *Trusted Computing, chapter 9,* 2005*, IEEE*.

[8] D. Dhillon, T. Randhawa, M. Wang, and L. Lamont, "Implementing a fully distributed certificate authority in an OLSR MANET," In *Proc. Wireless Communications and Networking Conference (IEEE WCNC 2004), 2004*, vol. 2, pp. 682–688.

[9] M. E. Elhdhili, « PKI hybride pour la gestion de clés dans les réseaux ad hoc,» Master Thesis, Ecole Nationale des Sciences de l'Informatique - Manouba, 2010.

[10] M. E. Elhdhili, L. B. Azzouz, and F. Kamoun, "A totally distributed cluster based key management model for ad hoc networks," In *Med-Hoc-Net 2004, The Third Annual Mediterranean Ad HocNetworking*, 2004.

[11] M. E. Elhdhili, L. B. Azzouz, and F. Kamoun, "Reputation based clustering algorithm for security management in ad hoc networks," *Int. J.Inf. Comput. Security*, 2009, vol.3, pp.228–244.

[12] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," In *Proc. 28th Annual Symp. on Foundations of Computer Science*, 1987, pp. 427–438.

[13] P.A. Fouque and J. Sern, "One round threshold discrete-log key generation without private channels," *Springer-Verlag*, PKC'01, 2001, LNCS, pp.300–316.

[14] Y. Frankel, P. Gemmell, P. D. MacKenzie, and M. Yung, "Proactive RSA," *In Proc. of CRYPTO 1997, the 17th Ann. Intl. Cryptology Conf*, 1997, pp. 440–454.

[15] Y. Frankel, P. D. MacKenzie, and M. Yung, "Robust efficient distributed rsa-key generation," In Proc. *PODC'98: Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, New York, 1998, pp. 320.

[16] Y. Fu, J. He, and G. Li, "A composite key management scheme for mobile ad hoc networks," In Proc. *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops*, Berlin, 2006, vol. 4277, pp. 575–584.

[17] S. Funabiki, T. Isohara, Y. Kitada, K. Takemori, and I. Sasase, "Public key management scheme with certificate management node for wireless ad hoc networks," In *Proc. International Multiconference on computer science and information technology*, 2006, pp. 445–451.

[18] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin, "Robust threshold DSS signatures," In *Proc. EUROCRYPT'96: the 15th annual international conference on Theory and application of cryptographic techniques*, Berlin, 1996, pp. 354–371.

[19] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," In *Proc. CRYPTO '95:Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, London, 1995, pp. 339–352.

[20] J.-P. Hubaux, T. Gross, J. Yves Le Boudec, and M. Vetterli, "Toward self-organized mobile ad hoc networks: The terminodes project," *IEEE Communications Magazine*, 2001.

[21] L. H. K. and C. Jaeyoung, “Multistage authentication scheme for mobile ad-hoc network using clustering mechanism,” *Lecture notes in computer science*, 2006, Vol.4208, pp.653–661.

[22] B. Kadri, A. M’hamed, and M. Feham, “Secured clustering algorithm for mobile ad hoc networks,” In *International Journal of Computer Science and Networks Security,* 2007, vol. 7, pp. 27–34.

[23] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, “Providing robust and ubiquitous security support for mobile ad-hoc networks,” In *Proc. Ninth Int Network Protocols Conf*, 2001, pp. 251–260.

[24] H. Luo and S. Lu, “Ubiquitous and robust authentication services for ad hoc wireless networks,” Technical Report, UCLA Computer Science Department, 2000.

[25] H. Luo, P. Zerfos, J. Kong, S. Lu, and L. Zhang, “Self-securing ad hoc wireless networks,” In *Proc. Seventh IEEE Symposium on Computers and Communications (ISCC)*, 2002.

[26] J. Luo, J P Hubaux, and P. T. Eugster, “Dictate: Distributed certification authority with probabilistic freshness for ad hoc networks,” *Trans. Dependable Secure Comput*, 2005, vol.2, pp.311–323.

[27] J. V. D. Merwe, D. Dawoud, and S. McDonald, “A survey on peer-to-peer key management for mobile ad hoc networks,” *ACM Comput. Surv.*, 2007, vol.39, pp.1.

[28] E. C. H. Ngai and M. R. Lyu, “Trust- and clustering-based authentication services in mobile ad hoc networks,” In *Proc. 24th International Conference on Distributed Computing Systems ICDCSW’04, Workshops - W7: EC (ICDCSW’04)*, Washington DC, 2004, pp. 582–587.

[29] P. Papadimitratos and Z. J. Haas, “Securing mobile ad hoc networks,” *Handbook of Ad Hoc Wireless Networks*. CRC Press, 2002.

[30] C. Park and K. Kurosawa, “New Elgamal type threshold digital signature scheme,” *IEICE transactions on fundamentals of electronics, communications and computer science*, 1996, vol.11 E79-A(1), pp.86–93.

[31] T. P. Pedersen, “A threshold cryptosystem without a trusted party,” In *Proc. EUROCRYPT’91: Proceedings of the 10th annual international conference on Theory and application of cryptographic techniques*, Berlin, 1991, pp. 522–526.

[32] T. P. Pedersen, “Non-interactive and information-theoretic secure verifiable secret sharing,” In *Proc. CRYPTO ’91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, London, UK, 1992, pp. 129–140.

[33] K. Ren, T. Li, Z. Wan, F. Bao, R. H. Deng, and K. Kim, “Highly reliable trust establishment scheme in ad hoc networks,” *Computer Networks*, 2004, vol.45, pp.687 – 699.

[34] G. Rosario, J. Stanislaw, K. Hugo, and R. Tal, “Secure distributed key generation for discrete-log based cryptosystems,” In *Proc. EUROCRYPT’99: Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, Berlin, 1999, pp. 295–310.

[35] Shamir, “How to share a secret,” *ACM 22*, 1979, vol.11, pp.612–613.

[36] V. Shoup, “Practical threshold signatures,” In *Proc EUROCRYPT’00: 19th international conference on Theory and application of cryptographic techniques*, Berlin, 2000, pp. 207–220.

[37] S. Čapkun, L. Buttyàn, and J.-P. Hubaux, “Self-organized public-key management for mobile ad hoc networks,” *IEEE Transactions on Mobile Computing*, 2003, vol.2, pp.52–64.

[38] S. Čapkun, J.-P. Hubaux, and L. Buttyàn, “Mobility helps peer-to-peer security,” *IEEE Transactions on Mobile Computing*, 2006, vol.5, pp.43–51.

[39] B. Wu, J. Wu, E. B. Fernandez, S. Magliveras, and M. Ilyas, “Secure and efficient key management in mobile ad hoc networks,” In *Proc. 19th IEEE Int. Parallel and Distributed Processing Symp*, London UK, 2005, vol. 30, pp. 937–954.

[40] S. Yi and R. Kravets, “Key management for heterogeneous ad hoc wireless networks,” In *Proc. ICNP ’02:10th IEEE International Conference on Network Protocols*, Washington DC, 2002, pp. 202–205.

[41] S. Yi and R. H. Kravets, “Composite key management for ad hoc networks,” In *Proc. Mobile and Ubiquitous Systems: Networking and Services, MOBIQUITOUS 2004, 2004.*

[42] S. Yi, and R. Kravets, “Moca: Mobile certificate authority for wireless ad hoc networks,” In *2nd Annual PKI Research Workshop Program (PKI 03)*, 2003, pp. 65–79.

[43] L. Zhou and Z. J. Haas, “Securing ad hoc networks,” *IEEE Network Magazine*, vol.13, pp.24–30, 1999.

[44] L. Zhou, F. B. Schneider, and R. van Renesse, “Coca: a secure distributed online certification authority,” In *Proc. [Organically Assured and Survivable Information Systems] Foundations of Intrusion Tolerant Systems*, 2003, pp. 152–191.

[45] B. Zhu, F. Bao, R. H. Deng, M. S. Kankanhalli, and G. Wang, “Efficient and robust key management for large mobile ad hoc networks,” *Computer Network*, 2005, vol. 48, pp.657–682.

[46] P. Zimmermann, “*The Official PGP User’s Guide*,” MIT Press Cambridge, MA, USA, 1995.

[47] C. Zouridaki, B. L. Mark, K. Gaj, and R. K. Thomas, “Distributed ca-based pki for mobile ad hoc networks using elliptic curve cryptography,” In *Proc. EuroPKI 2004,* 2004, *vol. 3093,* pp. 232–245.

Hella Kaffel-Ben Ayed received both engineering degree and Ph.D degree from the Faculty of Science of Tunis University of Tunis El Manar in 1989. From 1989 to 1993 she served as an engineer at Centre de Calcul El Khawarizmi. From 1984 to 1989 she was an assistant then assistant professor since 1993 at the Faculty of Sciences of Tunis, teaching graduate and undergraduate courses in computer networks, e-commerce, and security. Her main research interests include communication protocol and security protocols for e-commerce, e-government as well as new mobile pervasive applications

.

Adel Belkhiri received his engineering degree in computer sciences in 2006 from the Faculty of Sciences of Tunis. He is preparing his Master Thesis at the same institution. He is a researcher at the CRISTAL lab. His research is focused on the establishment of security over ad hoc networks.