SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks. 2010; **3**:150–166
Published online 20 May 2009 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.116

A secure and resistant architecture against attacks for mobile ad hoc networks

Abderrezak Rachedi*,† and Abderrahim Benslimane
LIA/CERI, University of Avignon, Agroparc BP 1228, 84911 Avignon, France

Summary

In this paper, we propose a new architecture based on an efficient trust model and secure distributed clustering algorithm (*SDCA*) in order to distribute a certification authority (*CA*) for ensuring the distribution of certificates in each cluster. We use the combination of a fully self-organized security for trust models like pretty good privacy (*PGP*) adapted to *ad hoc* technology and the clustering algorithm which is based on the use of trust and mobility metrics, in order to select the clusterhead and to establish a public key infrastructure (*PKI*) in each cluster for authentication and exchange of data. Furthermore, we present a new approach: the dynamic demilitarized zone (*DDMZ*) to protect the CA in each cluster. The principal idea of *DDMZ* consists in selecting the dispensable nodes, also called registration authorities (*RAs*); these nodes must be confident and located at one-hope from the *CA*. Their roles are to receive, filter and treat the requests from any unknown node to the *CA*. With this approach, we can avoid the single point of failure in each cluster. Moreover, we propose a probabilistic model to define the direct connectivity between confident nodes in order to study the resistance degree of the *DDMZ* against different attacks. In addition, we evaluate the performance of the proposed *SDCA* and we estimate the robustness and the availability of *DDMZ* through the simulations. The effects of direct connectivity and transmission range on the stability and security of the network are analyzed. The simulation's results confirm that the proposed architecture is scalable, secure, and more resistant against attacks. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: MANET; security; distributed *PKI*; trust model; clustering algorithm

1. Introduction

A mobile *ad hoc* network (MANET) is formed by a group of autonomous nodes without any pre-existence of a fixed network infrastructure. These nodes communicate with each other through wireless links, if the nodes are not within proximity, they can communicate with each other by using multi-hop links to forward traffic from the source to the destination. The nodes' mobility results in a dynamically changing network topology. All these characteristics make MANET original and popular in different application fields such as rescue missions, military operations, etc. However, the characteristics of MANET make it vulnerable to several attacks. In addition, a lack of central administration entity

*Correspondence to: Abderrezak Rachedi, University of Avignon, Agroparc BP 1228, 84911 Avignon, France.
†E-mail: abderrezak.rachedi@univ-avignon.fr

in MANET makes the existing security solutions not applicable to MANET. Hence, providing secure solutions for MANET is a challenging task. The goal of secure solutions is to provide security services, such as authentication, confidentiality, integrity, no-repudiation, and availability to mobile users. In order to achieve these goals, some questions need to be solved, e.g., (1) who ensures the security services in MANET? (2) Which are the characteristics of the nodes that are capable of ensuring the security services? etc.

Security schemes for MANETs usually are based on one or several of the following cryptography approaches: symmetric cryptography, public key cryptography (digital certificates) or threshold cryptography. Each approach has its advantages and drawbacks. For example, to secure an on-demand routing protocol, Perrig and Johnson [1] proposed the *Ariadne* protocol, a secure version of the dynamic source routing (DSR) [2] through using the broadcast authentication protocol *TESLA* (timed efficient stream loss-tolerant authentication) [3]. The authors used on symmetric cryptography algorithms and a one-way hash chain to establish the secure communication between two nodes. However, no trust model is proposed in the *Ariadne* protocol, and it is less robust and offers a lower degree of security than those involving asymmetric key cryptography. Sinzgiri *et al.* Reference [4] presented the authenticated routing in *ad hoc* networks protocol (*ARAN*), a secure version of the AODV protocol [5], by using a single certification authority (*CA*) for the whole network. The trust model in ARAN is based on a central *CA* which is not suitable in MANET. In order to overcome the problem of the central *CA*, Zhou and Haas [6] proposed the distribution of the *CA* by using the threshold cryptography scheme. The idea consists in decentralizing the *CA* functionalities over some specific nodes. Other works use a distributed CA [7] which is based on Reference [6]. The concept of threshold cryptography usually called (k, n) threshold scheme consists in splitting the secret to n partial shares, so the combination of at least certain k components among n $(k < n)$ can reconstitute the secret, otherwise the reconstruction of the secret is not possible. Other schemes are based on the threshold cryptography with the clusters concept, like Bechler *et al.*'s architecture [8]. In this architecture, the authors assumed that clusterhead nodes were confident and that they had a part of the *CA*'s secret. However, this architecture is not self-organized, and it needs the external administration to organize the clusters and to select the clusterheads when the topology changes.

In this paper, we propose a new architecture based on an efficient trust model and distributed clustering algorithm for designing the specific public key management system adapted to MANET's environment. We present the self-organized public key infrastructure (*PKI*) based on the distributed trust model and the clustering concept. The proposed trust model is adapted to a dynamic environment, because it can work as pretty good privacy (*PGP*) trust model [9] adapted to MANET's characteristics, like the fully self-organized security proposed by Hubaux *et al.* in Reference [10]. In addition, the trust model can work as a hierarchical model in the clusters. We propose the secure distributed clustering algorithm (*SDCA*) to elect the *CA* node in each cluster. The *SDCA* uses the trust level and mobility metric to elect the cluster head (*CH*) which becomes the *CA* in the cluster. In order to secure the *CA* and to avoid the single point failure in the cluster, we propose a new scheme which uses dispensable confident nodes, called registration authorities (*RA*). It consists of providing a dynamic demilitarized zone (*DDMZ*) at one-hope from the *CA* in each cluster. The role of *RAs* is to protect the *CA*, by receiving requests of certification, filtering, and treating these demands before forwarding them to the *CA*. Furthermore, a probabilistic model is proposed to study the connectivity between confident nodes and to evaluate the robustness of the proposed architecture. We evaluate the performance of the proposed *SDCA*, and we focus on the robustness and availability of the *DDMZ*.

The rest of the paper is organized as follows. In Section 2, we discuss the related works on current key management systems and the distributed CA approach with and without a clustering concept developed for MANET. In Section 3, we describe the proposed distributed architecture, our trust model and the *SDCA*. In Section 4, we present the monitoring and the cluster management design in order to dynamically update the trust model. In Section 5, we present our confident connectivity model to model the connectivity of the confident community. This model enables to evaluate the probability to form the secure cluster with the *DDMZ* robustness. In Section 6, we present the theoretical results of our confident connectivity model and the simulation results of the proposed distributed hierarchical architecture in order to study the network stability and the clusters' robustness. The Section 7 presents the comparison between our proposed architecture and others' architectures and protocols. The Section 8 is devoted to the security analysis of the proposed architecture and the quality

DOI: 10.1002/sec

of the authentication (QoA) is discussed. Finally, Section 9 concludes the paper and present our future works.

2. Related Works

PKI [11] consists of ensuring the communication security by providing data integrity, confidentiality, strong authentication, and non-repudiation. *PKI* is based on asymmetric cryptography algorithms and trust third parties (*TTPs*) called *CAs*. The *CA*'s role is to sign the nodes' public key before their authentication. In addition, the *CA* attributes a certain trust level to each node with a certificate. The certificates enable to establish different trust relationships among the *PKI*'s members. In contrast, *PKI* cannot directly be applied to MANETs because it is designed for centralized and well-connected networks. We know that among MANETs' characteristics we can quote the lack of central control entities, the dynamic network topology and resources can change frequently, so these characteristics complicate the *PKI* application in MANETs. The drawbacks of *PKI* in MANETs are as follows: select a single mobile node as *CA* for the entire network will create a single point of failure; if the *CA* is compromised, the entire network becomes compromised. In addition, the network's security is not scalable and the *CA* cannot be reachable by all nodes due to the nodes' mobility. Moreover, duplicate the *CA* in the network can improve the availability of the security services, but cannot eliminate the vulnerability. If only one of the *CAs* is compromised, the entire set of other *CAs* becomes compromised. Several works tackled the *PKI* problem in MANET, and most of them overcome the drawbacks of *PKI* in MANET by decentralizing the *CA* functionalities and self-organized *PKI*. We distinguish two main classes of the decentralized certification authorities in MANET. We called the first non-self-organized *PKI* which consists in distributing the *CA*' functionalities among several nodes by using the threshold cryptography. The second is self-organized *PKI* based on the trust model like *PGP* or others [9].

2.1. Distributed CA Approach Using Threshold Cryptography

The main idea consists in distributing the *CA* functionalities (not the *CA* duplication) among several nodes in the network by using the threshold cryptography scheme. In this scheme, the secrete key of *CA* is divided into n partial secretes $(S_1, S_2, \cdots, S_n)$ and each partial secrete is attributed to a certain node. However, only nodes that have a partial secrete are able to generate the partial certificate to the requested node. If the node wants to obtain a certificate, it needs to collect at least k $(k < n)$ partial certificates from different nodes which have a partial secret of the *CA* for generating the final certificate. So this scheme is called (n, k) threshold cryptography. The existing threshold cryptosystems are based on Shamir's (k, n) secret sharing [12]. In order to distinguish many distributed *CA* approaches, we focus on the following question: which nodes are able to sign the nodes' public keys and to generate a partial network certificate? On the other hand, who can obtain the partial secrete of the *CA* role? There are two possibilities of *CA* distribution: the first is called 'partial *CA* distribution', only certain specific nodes are able to play the *CA*'s role. The second is called 'full *CA* distribution'. In this case, all member nodes are able to play the *CA* role.

2.1.1. Partial CA distribution

For security reasons, the CA role distribution is limited to certain nodes. We can quote some works such as: Budakolglu and Gulliver [13] system based on the distribution of the CA among specific nodes with several threshold levels to offer nodes flexibility in selecting an appropriate security level for a given application. With this approach the fault tolerant and hierarchical key management services are ensured. In other works, Yi and Kravets [14] proposed the mobile certification authority called MOCA. In this approach, a mobile node which is more secure is selected to provide the *CA* functionality. MOCA nodes use the threshold cryptography to share their functionality and to increase the availability in the network. When a node wants to join a network, it sends at least k Certification Requests (CREQs) to different MOCA nodes and it waits for k replies from MOCA nodes. Each MOCA node which receives a CREQ will send the certification reply (CREP) which contains the partial certificate. The node obtains its final certificate signed by the *CA* when it receives at least k partial certificates from different MOCA nodes. Dong *et al.* [15] propose the distribution of the *CA* service by using threshold cryptography and they introduce the cluster structure. The cluster concept is adopted to provide the *CA* service and proactive secret shared update protocol. Another work based on the cluster architecture concept to distribute the *CA* node is proposed by Bechler *et al.* [8]. This architecture uses the threshold cryptography scheme

DOI: 10.1002/sec

(n, k) to distribute the *CA*. The idea is to distribute the private key of *CA* over *CHs* where every *CH* holds a fragment of the whole key. In order to be certified, any guest node must possess a certain number (W) of warranty certificates from warrantor nodes. Then, it must request at least (k) certificates from different *CHs*, whose association gives the network certificate. The drawbacks of this architecture are numerous: first, this approach is not realistic, because the warrantors do not have any information about the new node to be guaranteed (the warrantors must have minimal information about the nodes, so that they can decide to guarantee or not). Secondly, even if the guest has already W certificates from guarantors, it cannot succeed to have K certificates from *CHs*, it will not be certified. Thirdly, the network traffic generated by each new node in this procedure is at least of $2(W + K)$ packets. Others drawbacks in merging networks process: it assembles several networks in one network. As the two network keys cannot be mixed, one of them must be dropped and the other must be distributed over the whole network. The criterion to choose the dominating key among these different network's keys depends on the number of *CHs* of each network. The network, which has the maximum number of *CHs*, will become the dominant network and its network key remains the *CA* private key. These processes present a point of failure, because in this architecture any node can form its own cluster. Thus, a set of malicious nodes can form their network with the maximum number of *CHs*, and then attack the network in order to merge in the network and take the *CA* control.

2.1.2. Full CA distribution

Kong *et al*. [16] proposed an approach based on the distribution of the *CA* private key into a coalition of nodes. Each coalition has at least k nodes, and they are located at 1 hop from each other. When a node wants to obtain its certificate, it only needs to send a local broadcast request to obtain the k certificate reply. If the node does not receive sufficient responses with some time limit, it should move to another location. The problem with this approach is the assumption of at least k neighbor nodes or more usually exit, so this assumption is judged as unrealistic and unreasonable.

Unfortunately, the non self-organized *PKI* schemes based on threshold cryptography have some drawbacks: first, the n nodes must be initialized by a trust authority which is responsible of the introduction of the partial secret of *CA* role. On the other hand, an external administration is necessary to configure the system and to establish the architecture. Secondly, the number k must be a trade-off between availability and robustness, it must be frequently updated. Thirdly, the system overloads the network because instead of sending only one request to obtain a certificate or revocation, the node must send at least k requests ($k - 1$ traffic add in network). As conclusion for this part, any proposed architecture must take into account the characteristics of MANETs.

2.2. Self-organized PKI

In security terms, we consider that a MANET is fully self-organized, meaning that there is no infrastructure, no central authority, no centralized trust third party, no central server, no secret shared, even in the initialization phase. *PGP* [9] is a famous self-organized *PKI* named 'web-of-trust'. The principle is that any user can sign another user's public key. The set of signatures form the network of trust relationships. *PGP* is designed to establish the trust relationship in the web but it is not designed for a fully self-organized network. The distribution nature of *PGP* seems favorable for MANET security schemes. However, the *PGP* is susceptible to intrusion of malicious nodes. For example, we suppose that a node A trusts another node B, if B is compromised then B can issue valid certificates to several other malicious nodes who would be implicitly trusted by A from B (transition of trust relationship).

Among self-organized *PKI* based on the *PGP* model adapted to MANETs context, we quote the Hubaux *et al*.'s [10] system. Certificates are stored and distributed by the users themselves, unlike in *PGP*, where this task is performed by on-line servers (certificate directories). In this system, each user maintains a local certificate repository. When two users want to check the public keys of each other, they merge their local certificate repositories to find appropriate certificate chains. The success of this approach depends on the construction of the local certificate repositories and the characteristics of the certificate graphs. The authors present two algorithms that users can use to build their local certificate repositories. The drawback of this approach is to assume that trust is transitive and the system becomes more vulnerable to the intrusion of malicious nodes. Satizabal *et al*. [17] proposed an extension of Hubaux *et al*.'s [10] work by simplifying the certification path discovery. The authors present a protocol to establish a virtual hierarchy among the nodes in the network based on peer-to-peer *PKI*. However, the main drawback with the hierarchical

DOI: 10.1002/sec

model is to create the single point failure at the node with a high level, so the compromise of this node's private key results in a compromise of the entire *PKI*.

In our distributed architecture, we propose the hybrid approach works as both hierarchical trust model in each cluster if the cluster is formed and as *PGP* model otherwise. We present the self-organized *PKI* based on the distributed trust model and the clustering concept. We introduced two new concepts: first, the confident community notion is defined as a set of nodes which have a high trust relationship between them. The role of the confident community is to organize and to establish a *PKI* in each cluster of the network. The second concept called *DDMZ* is formed by a set of confident nodes located at 1 hop from the elected *CA* node in the cluster. Thus, the main goal of the *DDMZ* is to prevent the single point failure in each cluster and to protect the *CA* node.

3. Architecture

Firstly, we define a new trust model on which our distributed architecture is based. Secondly, we present a clustering algorithm-based trust and mobility metric to ensure a selection of trust and relatively stable confident node as *CH* which will become *CA* node in the cluster. Finally, we discuss how to evaluate certificate chain between clusters.

3.1. Primitives

The basic idea of our architecture consists of establishing a dynamic public key infrastructure with a *CA* as clusterhead that will change according to topology changes. We propose a *SDCA* based on this trust model. A new concept called *DDMZ* is proposed to protect the *CA* node in each cluster based on dispensable nodes.

Definition 1. *DDMZ is defined as the zone at* $1hop$ *from the CA. It is formed by at least one or more confident nodes called the RA. Their role is not to authorize unknown nodes to communicate directly with the CA node. All guest nodes must be passed by the DDMZ to request a certificate from the CA.*

We assume that there are spare social relationships among nodes in order to establish a trust relationship between any to-be-trust node and confident nodes. The set of confident nodes in the network is called 'confident community'. Every node also has its own private/public key pair. Furthermore, the initial trust nodes (or confident nodes) are honest and do not issue false certificates. Moreover, each node manages a trust table. Initially, each trust node knows the identity and public key of other trust nodes (ID, $K+$). It means, if we initially have k trust nodes, these nodes have $k-1$ entries (mutually known) in their trust table.

3.2. Trust Model

The trust model which defines the trust metric (Tm) and attributes to each role a certain trust degree: according to the node's trust level, a node can occupy a service function in the network. The trust level is represented by the trust metric which is a continue value into [0–1]. A node i has a high trust value ($\mathrm{Tm}(i) = 1$), if it is known and if it exchanged keys over a secure side channel (physical encounters and friends) with one or more several confident nodes [10]. Another manner to obtain a high trust value is that a node must prove its good faith by adapting a good behavior and cooperation. Each new unknown node starts with $\mathrm{Tm} = 0.1$ its lowest trust level.

Five roles of nodes are defined in each cluster and each role has a particular trust level:

- CA_k: CA of cluster k with certificate public key of nodes belonging to the same cluster. CA_k has a high trust level, Tm value must be equal to one.
- $\mathrm{RA}_{i,k}$: RA of cluster k assured by trust node i. The main goal of *RA* is to protect the *CA* against attackers by the *DDMZ* forming in order to avoid direct communication between unknown nodes and the *CA*, for example, they treat and filter the requests of the certification toward the *CA*. *RAs* must also be confident nodes with $\mathrm{Tm}(i) = 1$.
- $\mathrm{GW}_{i,j}$: It is a gateway node ensuring a connection between two different clusters i and j. These nodes must be certified by two different *CAs*. *GW* nodes must have a good trust level with $\mathrm{Tm}(g) \in [0.7–1.0]$.
- $\mathrm{MN}_{i,k}$: it represents a member node i belonging to the cluster k which successes to pass from visitor to member status with their good behavior and good cooperation. This status can be recommended by CA_k to another *CA* node. Node i can communicate inside and outside its cluster. It has an average trust level $\mathrm{Tm}(i) \in [0.5–0.7]$.
- $\mathrm{VN}_{i,k}$: It is a visitor node i that belongs to the cluster k, it has a low trust certificate, because CA_k and $\mathrm{RA}_{j,k}$ nodes need to have more information about node i's behavior. Node i cannot communicate outside its cluster. It has a minimal trust level $\mathrm{Tm}(i) \in [0.1–0.5]$.

DOI: 10.1002/sec

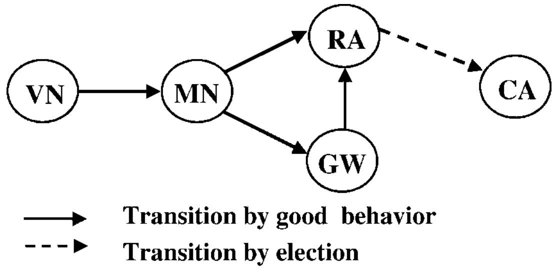

Fig. 1. State transition diagram.

Figure 1 shows the state transition diagram where each state represents the node's role in each cluster.

In our trust model, the trust relationship is ensured by *CAs* between clusters. A *CA* can recommend nodes with a certain trust level, belonging to its cluster to another *CA*. It is also ensured by *RA* in order to recommend nodes which belong to the same cluster to the *CA*.

The trust value of a path depends on its trust chain which is represented by its certificate chain. The inter-cluster communication is based on the evaluation of the certificate chain. The trust evaluation between two nodes consists of taking the smallest trust value of nodes (e.g., trust between *RA* and *GW* is min(1, w) where $w \in [0.7–0.9]$). Figure 2 shows two examples of certificate chains. The best trust chain (*TC*) is given in the case of b: TC(b) > TC(a).

In the case of a 1 hop cluster size, the *Tm* of the *GW* node must be equal to 1, in order to ensure the protection of the *CA* node by playing the *RA*'s role.

A trust *ad hoc* network can be represented like an undirected graph G = (V, E) where a vertex v belonging to V represents the *ID* and its corresponding public key and an edge e belonging to E represents the certificate. A directed edge from node i to node j will exist if there is a certificate signed with the private key of node i (K_i-) that binds the identifier of node j (ID_j) and its public key (K_j+). Figure 3 shows a simple example of trust *ad hoc* network which is constituted of nine nodes ($\|V\| = 9$) with {1, 2, 3} as the set of confident nodes which form a complete graph. Any confident node can become *CA* or *RA*. The *VN* status is attributed to unknown nodes in order to prove their well behaviors.

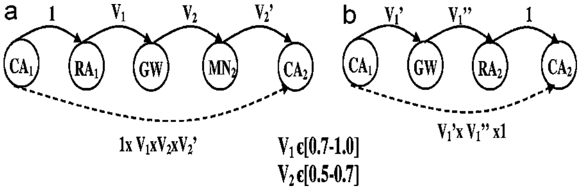

Fig. 2. Certificate chain.

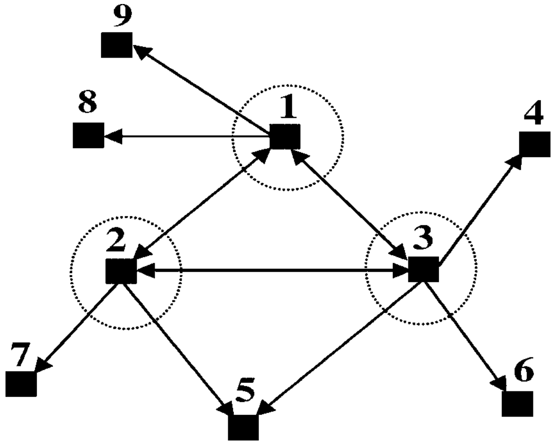

Fig. 3. A certification graph.

3.3. Secure Distributed Clustering Algorithm

A clustering network in our architecture is ensured by a *SDCA*. In *SDCA*, we select a *CH* which become the *CA* node, according to two criteria. First, the security is related to the trust level of a *CA* candidate and its number of confident neighbor nodes. Secondly, we use the mobility metric to have more stability in the cluster. Furthermore, *SDCA* ensures the authentication and integrity of the DATA in the election packet.

The main rules of this algorithm (*SDCA*) are the following:

1. Only confident nodes (Tm(i) = 1) can be the candidate to become CA.
2. Each cluster head is the CA of only one cluster.
3. All confident neighbors of the CA can become RA in the cluster.
4. Other nodes are at a distance of maximum d hop from the CA according the predefined size of cluster.

Our algorithm selects the CA of the cluster according to a trade-off between security and stability. It is based on sending periodic beacons by each confident node to its neighbors at a predefined interval time. Based on the information available in the received beacons and after authentication and verification of the beacon's integrity, the receivers update their information and decide about their cluster status.

The security parameter depends on the trust metric, only nodes with Tm = 1 and at least one trust neighbor (to establish DDMZ) can be candidate to become CA in the cluster. This constitutes the cluster formation condition. Moreover, to reinforce the security of the cluster, the algorithm selects the candidate with maximum trust degree; its means the trust node which has a maximum trust neighbors.

The stability parameter is very important on clustering algorithm, this parameter is defined as cluster head duration. Several clustering strategies have been proposed in order to increase system stability, such as: lowest-ID cluster head selection based on ID [18], max-connectivity algorithm [19]. In our algorithm, we adopt a mobility metric [20,21] because this strategy gives good result 33% of reduction in the number of cluster head changes compared with last approach [18,19].

Each trust node puts the following information in the beacon before transmission:

- *CA (cluster head)*: ID of the CA to which the node is attached.
- *HopCount*: hop count number to CA.
- *Degree of trust neighbors (DTN)*: each transmitter puts the number of its trust neighbors and their identities.
- *Relative mobility (RM)*: it indicates the relative stability of the trust node with respect to its trust neighbors as presented in Reference [22].
- *ID number of the beacon (ID-num)*: it is the sequence number of the beacon which is incremented by one at each beacon transmission by the node.
- *Message authenticated code (MAC)*: this field is reserved to authenticate the beacon information signed with private key of the sender.

$$\text{MAC} = E_{K^-}(H(\text{CA, Hopcount, DTN, RM, ID} - \text{num}) \qquad m)$$

This information permits any trust receiver to authenticate the sender of the beacon and verify the integrity of information.

At first, each trust node sends successive hello packet in order to calculate the relative mobility RM, after that, it announces itself as CA by assigning its own address to the CA field of the election beacon and initializes the hop count. When a trust node receives a beacon, from one of its neighbors, it executes the clustering algorithm to change its status from clusterhead (CH that is also CA) to RA or cluster-member only. The decision to change the status from CA to cluster-member depends on two main factors: security and stability parameters. Two security parameters in this algorithm have been defined: the trust level and the numbers of trust neighbors of CA candidate. These parameters indicate the security hardiness of the future cluster and the degree of attacks resistance. Another parameter is introduced: the stability of the CA. A CA is considered more stable than another one if it has low relative mobility. Any trust node with relative mobility more than certain threshold is not considered stable and will not enter in CA competition with others candidates. When competition between two candidate CAs, the CA with lower trust neighbors and also more relative mobility, loses the competition and becomes RA or member only, it depends on the distance (i.e., hop count) from the winner CA. If the hop count is equal to 1, the candidate CA becomes RA. It means that all trust nodes, directly connected (1 hop) to CA winner, can become RA. The nodes situated between two adjacent clusters can become gateway (GW). The below Algorithm 1 is executed by each node which has high trust metric Tm = 1; these nodes declare themselves as candidate to become CA. The extent of a node CA to manage nodes (in its cluster) at 1 hope or more depends on the value of *k* which depends on the cluster size.

Algorithm 1: Clustering Algorithm executed by confident node

```
When node j receives a beacon from node i;
begin
    Authentication do if $(Tm(i)! = 1)$ then
    RejectBeacon(); Goto(end);
    else if $(HopCount >= k)$ then
        No − Competition; Goto(end);
    else if $(RM_i < RM_j)$ OR $((RM_i == RM_j)$
    AND $(DTN_j < DTN_i))$ then
        Accept node i as CA;
        if $(HopCount == 1)$ then
            $Status(j) = RA$;
            $HopCount(i) = 1$;
        else
            $HopCount(i) = HopCount + 1$;
            $Status(j) = MN$;
    else if $(RM_j < RM_i)$ OR $(DTN_j > DTN_i)$
    then
        node j remains as CA candidate;
    else if $(RM_i == RM_j)$ AND
    $(DTN_j == DTN_i)$ then
        apply Lowest-ID;
end
```

In order to detect the topology changes, we introduce the movement detection process. Movement of CA is detected by RA nodes while not receiving any

DOI: 10.1002/sec

beacon from CA for predefined period of time. Also, cluster's nodes can detect movement of RA nodes by not receiving beacons from them. The movement detection of nodes CA and RA is very important for the cluster lifetime.

Algorithm 2: Algorithm executed by a node when its RA or CA is lost

```
If node i does not receive any beacon from CA
after Timeout predefined;
begin
    if It can recover CA with another RA then
        Keep previous CA;
        Update RA node and $Hop\_count$;
    else if It can find another CA then
        Join the new CA node;
        if ($Tm(i) == 1$) then
            if ($HopCount == 1$) then
                $Status(i) = RA\_NODE$;
                $HopCount(newCA) = 1$;
            else
                $Status(i) = MN$;
                $HopCount(newCA) =$
                $HopCount + 1$;
        else
            Request Certificate to RA node;
end
```

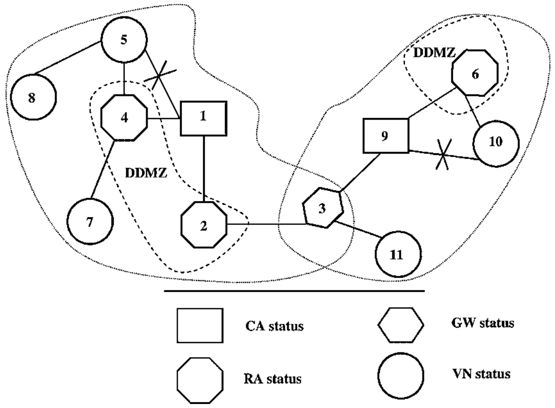

Fig. 4. An example of 2 hope cluster formation.

Each cluster's node other than CA or RA receives the beacon from CA. It must verify the authentication and the integrity of the beacon information by using the CA's public key ($K_{CA}+$). If the verification succeeds then the node updates any change about hop-count or new RA. If CA changes, cluster nodes verify the new CA identity. The information over the nodes can be assembled by trust model.

Each member cluster's node update periodically the cluster's information (CA and RA nodes), for more detail, the reader can refer to Reference [22]. Figure 4 shows an example of the network clustering result by applying our algorithm, in the case of 2 hop of cluster size. As shown in this figure, nodes 2, 4, 5, 7, 8, and 3 are nodes of the cluster within the clusterhead 1 as CA and 3, 6, 10, and 11 are nodes of the cluster within the clusterhead 9 as CA. Nodes 2 and 4 are confident nodes and they are one hope from CA. In this case, they have RA status. This is also the case for node 6 in the second cluster. Node 3 belongs to both clusters, it can become gateway if it has certain trust value. The other nodes 5, 8, 7, 10, and 11 have initially visitor status and cannot communicate directly with CA node despite nodes 5 and 10 are neighboring from CA node.

4. Monitoring and Cluster Management

In this section, we present the design of the monitoring and the cluster manager modules.

4.1. Monitoring

The hierarchical monitoring process consists of supervise behaviors of nodes. Each node with high trust value monitors its neighbors' nodes with low or equal trust values. Figure 5 shows the possibility of a node with certain status to monitor other status nodes. CA can monitor other CAs and all other status. RA can monitor {CA,GW,MN,VN} status, also GW can supervise {GW,MN,VN} status. Finally, MN node can supervise {MN,VN} status but VN can supervise only the VN nodes.

In the monitoring module, each node with a high trust level monitors its neighbor nodes with a low

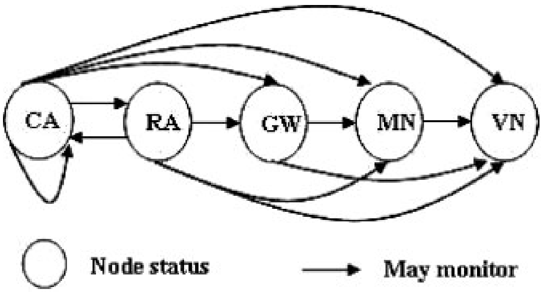

Fig. 5. Monitoring scheme.

DOI: 10.1002/sec

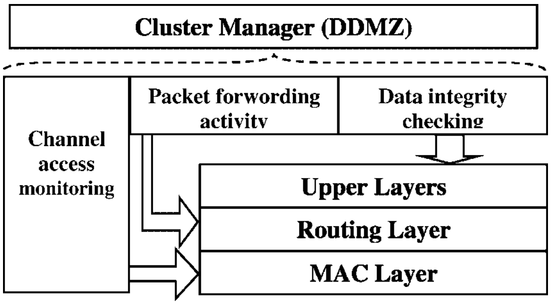

Fig. 6. The different component of the monitoring module.

trust level. The monitor process is needful to ensure the dynamic trust level updating. In order to ensure this task, the monitor process in our case acts in two different network protocol layers: MAC layer and network layer. Figure 6 shows the different components of the monitoring module and the interaction with the cluster manager. The monitoring module intervenes in different protocols layers:

- *MAC Layer*: the monitoring nodes supervise the channel occupation by their neighbors. This consists of measuring the duration of the channel occupation by nodes with their own flows. This function is motivated by one type of selfish misbehavior proposed by Reference [23], that is selfish nodes cheat from the choice of backoff in order to access more bandwidth than well-behaved nodes. Several solutions have been proposed in literature to monitor the selfish nodes in MAC layer. Hubaux *et al.* Reference [24] proposed DOMINO, a system to detect the cheating behavior in WLANs, however, DOMINO can be exploited by several smart selfish behavior. Guang *et al.* [25] proposed a new algorithm, predictable random backoff (PRB), to prevent against these attacks, this solution needs to modify of IEEE 802.11 binary exponential backoff (BEB) and forces each node to generate PRB intervals. So, each monitor node generates a report (R_1) about their neighbors with a low trust metric. In this paper, we do not focus in the MAC layer monitoring.
- *Network layer*: the monitoring nodes supervise the packet forwarding activities of its neighbor nodes with low trust metric. The idea is based on the proportion of correctly forwarded packets with respect to the total number of packets to be forwarded during a fixed time interval (monitoring period). The monitoring period is the observation period which consists in collecting the information from nodes in order to calculate the reputation rating. Let node x and y with $\mathrm{Tm}(x) > \mathrm{Tm}(y)$. In this case, the node x can monitor the node y. The node x sends a certain number of packets to the node y with an other destination node, after a fixed time interval, the node x can calculate the reputation rating:

$$\mathrm{R}_2(X, Y) = \frac{\text{Number of forwarded packets}}{\text{Number of sent packets}} \quad (1)$$

Yacine *et al.* [26] proposed a similar idea to calculate the reputation rating, the difference between our monitoring module and Yacine's module is the attribution of trust metric. In our case, each unknown node starts with a low trust metric ($\mathrm{Tm} = 0.1$) and increases when it proves its cooperation and well behavior (has a good reputation rating), whereas in Yacine's monitoring module all nodes start with a high trust metric ($\mathrm{Tm} = 1$) and decrease when they misbehave. The problem with this approach is that the malicious nodes can profit from it and generate a false reputation report for nodes which are well behaved. In our approach, we take into account the trust metric of the monitor nodes. The reputation ratings generated by nodes are related to the trust metrics corresponding to each node. It is the task of the cluster manager. The final MAC and routing layers report about node y generated by each monitor node x is

$$\mathrm{R}(x, y) = \frac{\mathrm{R}_1(x, y) + \mathrm{R}_2(x, y)}{2} \quad (2)$$

4.2. Cluster Manager

The cluster manager is formed by the CA node and a set of RA nodes with high trust levels. The role of the cluster manager is to ensure the cluster security where the CA node will generate a certificate for cluster members. A set of RA nodes forms the DDMZ in order to protect the CA node against attacks *via* filtering communications from any unknown node to the CA node. The DDMZ use the reputation rating from the monitoring process to evaluate the cluster members.

The cluster manager module collects the reputation report from the cluster members. The monitor nodes generate on-demand reputation rating reports. The cluster manager enquries the monitor nodes to generate the report of certain nodes. When the cluster manager receives the reputation rating report from the monitor

DOI: 10.1002/sec

nodes, the aggregation component is executed. If the cluster manager receives k reports from the monitor nodes x_i to evaluate the node y, then

$$\text{Reputation Report:RR}(y) = \frac{1}{k}\sum_{i=1}^{k}\text{Tm}(x_i) \times \text{R}(x_i, y) \tag{3}$$

When the cluster manager has the reputation reports, the behavior classification is executed to classify the nodes. If the reputation report exceeds a certain threshold th_1, the trust metric increases otherwise the trust metric does not change if the reputation report is between th_0 and th_1 ($\text{th}_0 < \text{th}_1$). The trust metric is decreased when the reputation report is less than th_0. However, if the report is negative, the trust metric becomes null and the misbehaved nodes will be punished. In the case, of a certain number of negative reports, the misbehaved nodes will be discarded from the cluster and the cluster manager informs the other adjacent cluster managers about recidivist-misbehaved nodes.

$$\begin{cases} \text{RR}(y) > \text{th}_1 & \text{Tm}(y) \text{ increases} \\ \text{th}_0 \le \text{RR}(y) \le \text{th}_1 & \text{Tm}(y) \text{ does not change} \\ \text{RR}(y) < \text{th}_0 & \text{Tm}(y) \text{ decreases} \\ \text{RR}(y) < 0 & \text{Tm}(y) = 0 \end{cases}$$

where, $\text{th}_1 > \text{th}_0 > 0$, these parameters can be defined by the cluster manager according to the cluster configuration and security level needed.

Figure 7 shows the different functions of the cluster manager and interaction with monitoring module.

Figure 8 shows the interaction between different modules, the monitoring, the election (SDCA) and the cluster manager modules, interact with a trust model as indicated in Figure 8 with 1, 2, 3 transition. The modules, election and cluster manager call the monitoring module to control the behaviors of the nodes (4, 6 transition). The cluster management module is the result of SDCA as indicated in Figure 8 with five transitions.

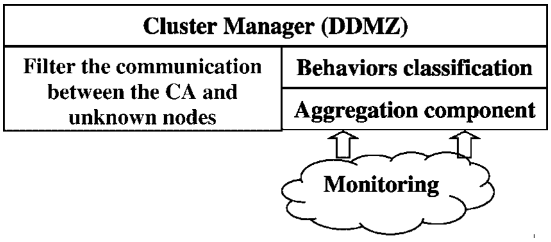

Fig. 7. Cluster manager functions.

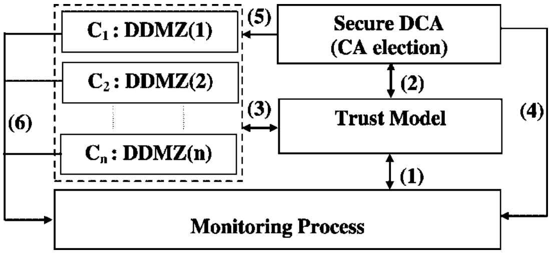

Fig. 8. Functional diagram for secure architecture.

In our distributed architecture, the nodes must belong to a cluster to be able to secure their communication with member nodes of the cluster, because nodes in the cluster communicate only with nodes which possess the certificate from the CA of this cluster. The communication between clusters is ensured by the gateway nodes which belong at least to two clusters and possess a certain trust level.

The security of our architecture depends directly on the trust model and connection degree between the confident nodes, because the connectivity of confident nodes indicates the availability of security services in the network, the degree of resistance against the attacks like denial of service (DoS), and the robustness of DDMZ in our architecture. For these reasons, we study the connectivity between confident nodes.

5. Confident Connectivity Model

The basic idea is to distribute k confident nodes among n total number of nodes in the network. These confident nodes must collaborate with each other in order to divide the network into different clusters and to assign the roles of CA and RA in each constructed cluster [27]. The conditions of the cluster's formation are: (a) the absence of any confident cluster which can accept other nodes (a cluster which is not saturated); (b) the existence of at least two confident nodes which must be directly connected. It is possible only if these two confident nodes are geographically close enough. In each cluster, the CA node and the confident nodes directly connected from the DDMZ.

We define direct communication as follows: two nodes (i) and (j) can directly communicate with each other if the node (i) belongs to the set of node (j)'s neighbors and if the node (j) belongs to the set of node

DOI: 10.1002/sec

(i)'s neighbors. If we assume that there is no obstacle in the area and that all nodes have the same transmission range R, we can formulate the direct connection by ($|Xi - Xj| < R$), i.e., X_i is the location of the node (i) [28].

We assume that n nodes are distributed with a Poisson arrival rate λ on R^+. The probability that any node (i) can directly communicate with any node (j) is

$$P(R) = \Pr\{|X_i - X_j| \leq R\} \tag{4}$$

As the inter-arrival distance of Poisson sequence is *iid* exponentially distributed and memoryless, we get the following probability:

$$P(R) = 1 - e^{-\lambda \times R} \tag{5}$$

In our case, the probability to have (d+*1*) confident nodes directly connected is

$$P_d(R) = \prod_{i=1}^{d}(1 - e^{-\lambda \times R}) = (1 - e^{-\lambda \times R})^d \tag{6}$$

The parameter d represents the degree of direct connectivity between confident nodes. Bhaskar *et al.* [29] showed that the probability to have a high connected network depends on the transmission range of the nodes. The higher the transmission range is, the greater the probability of a connected network is.

The probability to get two nodes (i) and (j) directly connected, knowing that they belong to the set of confident community K which contains $|K| = k$ confident nodes in the network of n total number of nodes (confident and not confident nodes) is

$$\begin{aligned} P &= P(R) \times \Pr\{\text{node}(i) \in K\} \\ &\quad \times \Pr\{\text{node}(j) \in K \setminus \text{node}(i) \in K\} \end{aligned} \tag{7}$$

$$P = P(R) \times \left(\frac{k}{n} \times \frac{k-1}{n-1}\right) \tag{8}$$

In the general case, the probability to get (d+*1*) nodes directly connected, knowing that they are confident node (their Tm = 1) is

$$P = P_d(R) \times \left\{\frac{k}{n} \times \frac{k-1}{n-1} \times \cdots \times \frac{k-d}{n-d}\right\} \tag{9}$$

where $d < k$ and $k \leq n$.

Finally, we get the probability of (d+*1*) confident nodes directly connected according to the transmission range R, the percentage of confident nodes in network (k/n) and the degree of direct connectivity d between confident nodes.

$$P = (1 - e^{-\lambda \times R})^d \times \left\{\frac{k}{n} \times \frac{k-1}{n-1} \times \cdots \times \frac{k-d}{n-d}\right\} \tag{10}$$

This probability shows the possibility to form the cluster in the network with (k/n) as rate of confident nodes.

In DDMZ, d is a parameter which indicates the robustness and the degree of resistance of DDMZ against attacks like DoS and also the availability of security services, like for instance, filtering CREQs before forwarding them to the CA node.

6. Performance Evaluation

6.1. Theoretical Results

In this subsection, we present the main simulation results of the confident connectivity model. In the following figures, we show the probability to get confident nodes directly connected as indicate in the Equation (10). Figure 9 shows the probability to get confident nodes directly connected with d degree according to the percentage of confident nodes in the network in the case of a high probability to get two nodes directly connected, that means in the case of a high transmission range ($P(R) = 0.9$). We notice that more the percentage of confident nodes increases, the probability to form the robustness of DDMZ increases

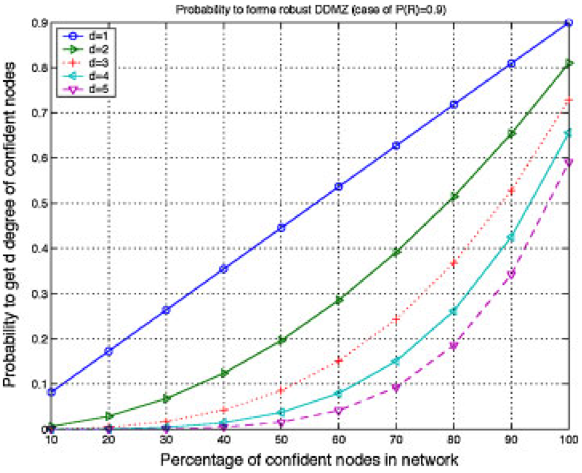

Fig. 9. Probability to form a DDMZ with degree d according to the confident nodes rate (P(R) = 0.9).

DOI: 10.1002/sec

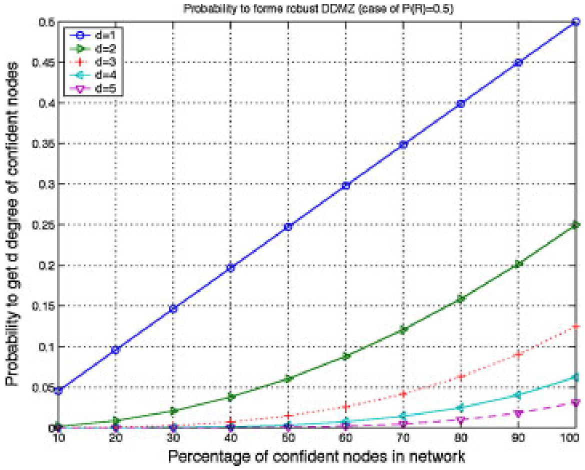

Figure 10. Probability to form a DDMZ with degree *d* according to the confident nodes rate (P(R) = 0.5).

with parameter (*d*). However, the probability to have any DDMZ with a small degree *d* is greater than the probability to have any DDMZ with a high degree *d*.

Figures 10 and 11 show the effects of the transmission range. In Figure 10, we plot the results in the case of the probability to get two nodes directly connected equal to 0.5 (P(R) = 0.5). We notice that the probability to get confident nodes directly connected with *d* degree according to the percentage of confident nodes in the network decreases compared to the case of a high probability of direct connectivity. We notice that the probability to form DDMZ decrease. In Figure 11, we illustrate the case of a low probability of direct connectivity ($P(R) = 0.2$). We notice that, the probability to get two confident nodes directly connected depends directly on the probability to get two nodes directly connected (P(R)). The P(R) depends on the transmission range, so we can conclude that the probability to get two confident nodes and the degree of connection between these nodes depends on the transmission range.

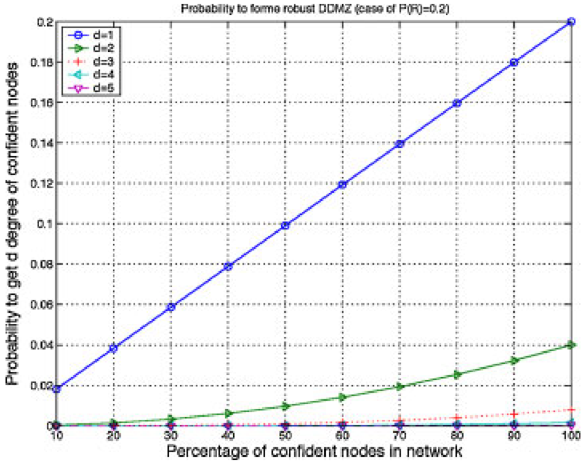

Fig. 11. Probability to form DDMZ with degree *d* according to the confident nodes rate (P(R) = 0.2).

Table I. Simulation parameters.

Parameter	Value in our simulation
Number of nodes (N)	50
Network size (mxn)	670 × 670 m^2
Mobility	[0–20 m/sec]
Transmission range	10–250 m
Pause time	0.30 s
Broadcast interval (BI)	0.75–1.25 s
Discovery interval	10× BI s
Contention period	3.0 s

6.2. Simulation Results

In this subsection, we evaluate the performance of the proposed secure clustering algorithm (SDCA) and the robustness of the proposed model. We have implemented our clustering algorithm as described previously. We use network simulator (NS-2) [30] with CMU wireless extensions to simulate our algorithm. Simulation scenarios were generated with parameters listed in the Table I.

The random waypoint model is selected as mobility model in the field (670 × 670 m^2) with the node speed uniformly distributed in [0–20 m\sec]. The total number of nodes in the network is $N = 50$.

6.2.1. SDCA performance

In order to compare the algorithm proposed in the previous section with others clustering algorithms. We assume that all nodes of the network are high trust level which means that any node can become CH.

In Figure 12, we note that there is difference between our algorithm, MOBIC and lowest-ID in the transmission range 50 m, because our algorithm need at least two nodes to form cluster, only one (isolated node) cannot become CA for security reason. In this simulation, the number of conceived clusters does not exceed 25. With a transmission range between 50 and 125 m the number of clusters quickly decreases and more of 150 m the network become more stable. However, while fixing the cluster size to the 2 hop,

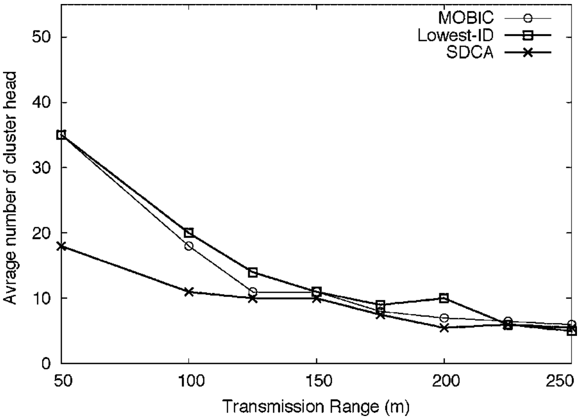

Fig. 12. Comparison between different clustering algorithms.

we obtain less clusterhead compared to MOBIC and lowest-ID.

Figure 13 shows the average number of different status of nodes in the network. The average number of isolated nodes (nodes cannot join any cluster) decreases when the transmission range increases. Also the number of CAs decreases with longer transmission range. The number of other nodes (member of different clusters) increases when the transmission range increases. The number of isolated nodes must be reduced to get more security communication in the network, because, any isolate node does not have a valid certification, so then, it cannot securely communicate.

The network overhead: The network overhead is an important parameter to evaluate the performance of the network and particularly the SDCA. Figure 14 shows the network overhead according to the simulation time. We remark that at the first 20 s the average network overhead is around 325 Ko. This important traffic due to the election process. However, between 20 and 40 s, we notice that the overhead decreases to 110 Ko and it stabilizes at 100 Ko until the end of simulation because the overhead restrict only to the clusters' beacon in order to maintain the clusters' formation.

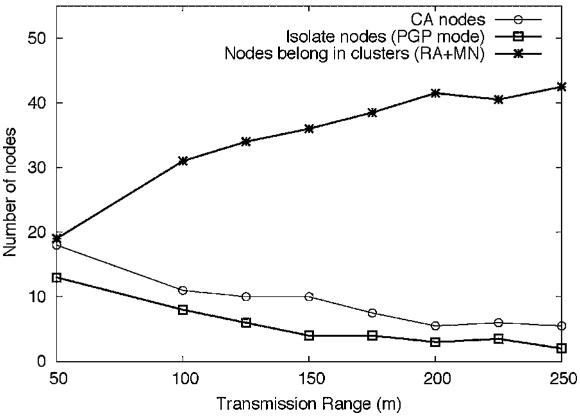

Fig. 13. Average number of different status of nodes.

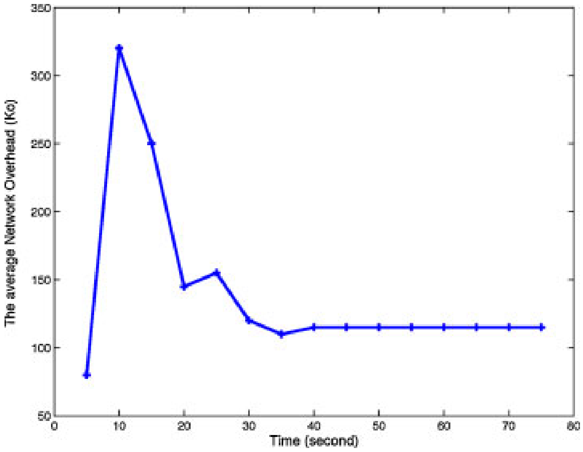

Fig. 14. The network overhead *versus* simulation time.

6.2.2. Percentage of confident nodes' impact

We study the impact of the confident nodes percentage on the proposed architecture. We plot in Figure 15 the average cluster number formed and the average *DDMZ* size in each cluster according to the percentage of confident nodes in the network. We notice that the number of the cluster formed increases when the percentage of confident nodes increases up to 30%. More than 30% of confident nodes the number of the

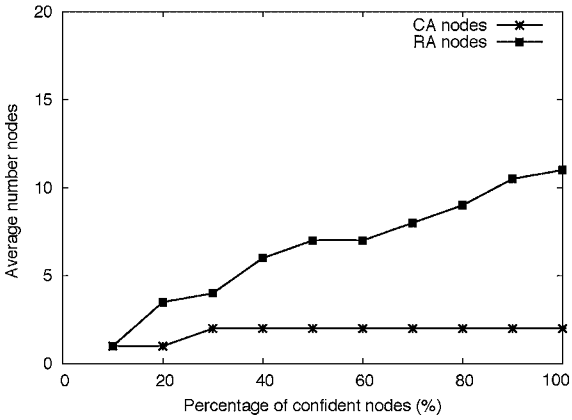

Fig. 15. The average CA and RA nodes versus the percentage of confident nodes in the network.

DOI: 10.1002/sec

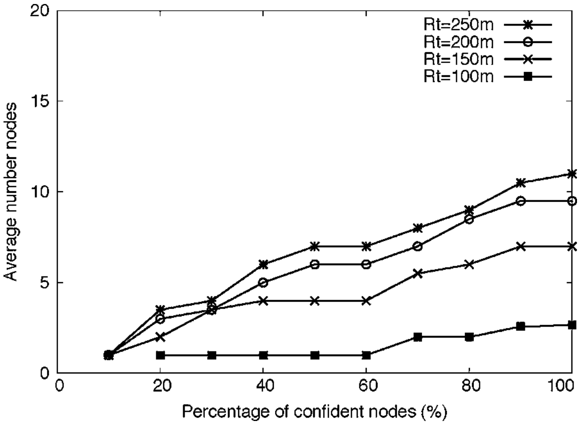

Fig. 16. The average *DDMZ* size *versus* the percentage of confident nodes in the network with different transmission range.

cluster stabilize around 3. However, the size of *DDMZ* linearly increases when the number of confident nodes increases. We can say that when the percentage of confident nodes reaches 30%, it does not affect the number of the cluster or the number of *CA* nodes.

In order to investigate the average *DDMZ* size (the number of *RA* in each cluster) with different transmission range, we plot in Figure 16 the average *DDMZ* size according to the percentage of confident nodes. We remark that when the transmission range equals to 250 m, the *DDMZ* increases linearly when the number of confident nodes increases. Once 50% of nodes in the network are confident, the average *DDMZ* size is around 6.5 of *RA* nodes but it reaches 11 of *RA* nodes when all nodes in the network are confident. However, when the transmission range decreases the average *DDMZ* size decreases too, but it significantly decreases when the percentage of confident nodes decreases. In the case of the transmission range equals to 100 m, we remark that the average *DDMZ* size cannot exceed 2.5 even all nodes in the network are confident.

We know that the *DDMZ* size has an important impact on the security cluster availability. We focus on the *DDMZ* availability because, it is an important parameter to evaluate the cluster robustness. This parameter is proportional to the number of *RA* nodes and the number of cluster member nodes. The *DDMZ* availability can be calculated according to the number of RA nodes and the total number of member nodes. The following formula defines the DDMZ availability of cluster i.

$$\text{Availability}(i) = \frac{\text{Number of RA nodes}}{\text{Total number of member nodes}}$$

In order to ensure the security services and to get a robust *DDMZ*, we need to maximize the *DDMZ* availability. The cluster with great member nodes needs more number of *RA* nodes (great *DDMZ* size) than cluster with low member nodes.

7. Comparative Study

This section is devoted to the comparison between our proposed architecture and others architectures and protocols. As comparison's metrics, we quoted

- *Self-configuration*: this metric is classified in two types: fully self-organized and Partial self-organized. The fully self-organized system does not need the central server, the secret shared, even in the initialization phase. On other hand, the system is able to organize itself without intervention of external administrator. However, the partial self-organized needs an intervention of external administrator either in the initialization phase or when the topology changes or others network configuration change.
- *Mobility support*: this metrics indicates if the mobility is taken into account in the scheme and the mobility impact on the trust model is considered. For example, when the CA becomes not reachable the security services must be ensured and an alternative is taken into account.
- *Reliability*: this metrics shows the ability of a system to perform its required functions under stated conditions for a specified period of time. The trust model must be dynamically updated and adapted to the environment change.
- *Scalability*: in security terms it indicates the ability of the system to keep an acceptable security level when the nodes' density increases. The robustness degree must be taken into account when the network size increases.
- *Flexibility*: shows the ability to easily adapt to different circumstances in MANET such as network topology change, security resources change, etc. This parameter is important in dynamic network like MANETs.
- *Availability* is the degree of a security services ensured by the system in the different situation change.
- *Energy consumption*: this metric indicates the cost needed by the scheme in energy consumption terms. We classify energy consumption of the system on tree levels: high, medium, and low energy consumption.

DOI: 10.1002/sec

Table II. Comparative table.

Metrics	DACA	MOCA [14]	BEC. [8]	HUB. [10]	BUD. [13]	SAT. [17]	DON. [15]
Self-configuration	✓	—	—	✓	—	✓	✓
Mobility support	✓	✓	—	✓	—	✓	✓
Reliability	✓	—	—	✓	✓	✓	✓
Scalability	✓	—	—	—	✓	—	✓
Flexibility	✓	—	—	✓	✓	✓	—
Energy consumption	Medium	High	High	Low	High	Low	High
Single point failure	—	—	—	—	—	—	—
First line of defence	✓	—	—	—	—	—	—
Trust updating	✓	—	✓	✓	—	✓	—
Fault tolerant	✓	—	—	—	✓	—	—

- *Single point failure*: this metric indicates the present or not of the single point failure in the system. It enables to evaluate the compromising risk on the CA.
- *First line of defence*: it shows if the first step to face the attack in the network exist or not. This metric permits to evaluate the robustness of the system.
- *Trust updating*: this metric enables to estimate the trust model dynamically refresh the nodes' trust. On other hand, the monitoring process is necessary to ensure trust model updating.
- *Fault tolerant*: this metrics indicates if the system continues to work, even some nodes are compromising. The compromising of any node does not disturb the entire network. On other hands, the system tolerates the existence of the malicious nodes and reduces the impact of their attacks.

The Table II illustrates the comparison between our proposal distributed architecture for certification authority (DACA) and others' architectures.

8. Security Analysis

The security of our architecture depends directly on the trust model. The presence of a great number of confident nodes increases the security of the network. Nodes with low trust level cannot participate in the CA election process. Only a confident node can announce itself as CA candidate. If a malicious node try to be introduced in the CA process election by announcing itself as candidate, the confident nodes can detect this in authentication phase showed in Algorithm 1. If malicious nodes succeed to form their cluster and try to communicate with other clusters; the CA of cluster destination can authenticate the CA of the source cluster in inter-cluster communication. All communications from a malicious cluster are ignored. The denial-of-service (DoS) attack over CA node is prevented by DDMZ where RA nodes filter all requests from unknown nodes. The robustness of DDMZ depends on the number of RAs which collaborate in order to protect CA of their cluster. If attackers try to impersonate legitimate nodes as CA or RA they will be detected by monitoring process and then isolated from the network. The malicious nodes can use the identity of legitimate nodes only if their private's keys are divulgated. If attackers try to compromise all the network, it must compromise all CAs.

The number of clusters formed by our proposed solution is related to the number and the mobility of confident nodes. The cluster size must be adapted with number of confident nodes in order to well secure CA node. The presence of two confident nodes is the minimum configuration of clustering and it must be reinforced.

We can use the thresholds cryptography scheme in each cluster after CA election. A CA divides its private key into n partial shares which are distributed over RA nodes. The advantage with this scheme is when the CA node is not available for any reason (mobility, lack of energy, ...) the cluster can continue to operate until the next election of another CA node.

Our system's architecture obliges nodes to collaborate and to adapt well behaviors to obtain more trust levels. Each unknown node must begin with a visitor status and then obtain the member status. To ensure the nodes' trust level evolution, a monitoring process is proposed to detect non-cooperative and malicious nodes.

In order to evaluate the trust of CA authentication, we calculate the QoA, so that, we apply attenuation factor

DOI: 10.1002/sec

to trust chain [31]. This factor is $(1 - p)^{(d-1)}$ where p is the probability of the existence of compromised or a malicious node in the network and d the length of the trust chain.

$$\mathrm{QoA}(V_1 - V_2) = \mathrm{TC}(V_1 - V_2) * (1 - p)^{(d-1)} \quad (11)$$

The more trust chain is longer the more risk to be compromised is important. In this case, the cluster size must be carefully chosen.

The QoA between two clusters depends of the trust chain (TC) which attach CA nodes of clusters and also percentage of malicious nodes in the network. The communication between CAs must passed via high trust chain and it is assured by GW nodes.

The Figure 17 illustrate the quality of authentication versus probability of malicious nodes. We have plot curves in the case of cluster size 1 and 2 hop with maximum and minimum values of TC respectively 1 and 0.49 (0.7 × 0.7). We remark the QoA linearly decrease with probability of malicious nodes increase in the case of one hop of cluster size. When we increase the cluster size at 2 hop we note that, QoA decrease more fast with probability of malicious nodes than the case of one hop of cluster size.

The Figure 18 shows the general case of QoA with different values of TC and probability of malicious nodes. We compare three cases of cluster sizes 1, 2, and 3 hop, we remark the best value of QoA is in the case of 1 hop of cluster size, low value of malicious nodes and high TC.

According to the last Figures 18 and 17, we can conclude that, the larger the cluster size is , the risk to have weak QoA is high.

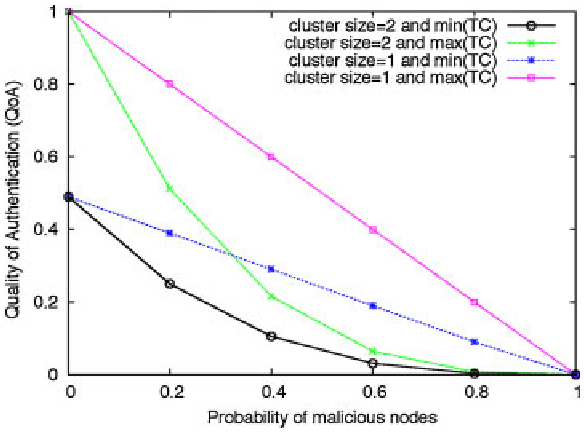

Fig. 17. QoA *versus* probability of malicious nodes.

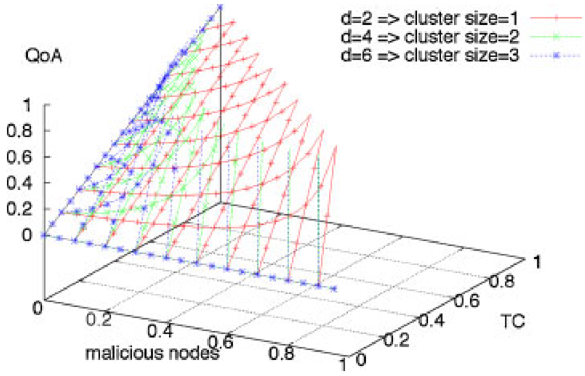

Fig. 18. QoA *versus* probability of malicious nodes and trust chain consideration.

9. Conclusion

In this paper, we have proposed a new architecture based on our trust model and secure clustering algorithm (SDCA) in order to distribute a CA. The SDCA is based on two parameters: security and stability. The security factor is related to the trust model; only confident nodes can become clusterhead and ensure CA role. The stability factor is presented by mobility metric in order to give more stable clusters. In our approach, the trust model is accomplished by monitoring process which allows any node with high trust metric to monitor and evaluate other nodes with low trust metric. In addition, we have proposed a new mechanism to protect CA, called DDMZ, which permits to increase security robustness of clusters and endures malicious nodes that try to attack CA or issue false certificates.

The secure proposed architecture ensures the security and availability of public key authentication in each cluster. This architecture is adapted to any topology changes. We proposed a confident connectivity model to study the security robustness in the clusters. We presented the different modules of our a secure distributed hierarchical architecture: the trust model, the election process, the cluster manager and the monitoring module. In this study, we focused on the cluster manager particularly on DDMZ, this approach consists on the protection of the CA in each cluster. The security of each cluster depends on the robustness and the availability of the RA which form the DDMZ. The DDMZ collaborate with monitoring module to develop the confident community. Each node with a low trust level needs to well-behaved to get a high trust level.

Simulation results of our clustering algorithm showed the improvement of clusters stability compared to MOBIC and lowest-ID algorithms. Furthermore, simulation results confirm our confident connectivity model, when the probability to get two nodes directly connected increases, the probability to get a robust DDMZ increases. We remark that availability and robustness of DDMZ depend on the transmission range, the number and mobility of confidant nodes. We are also considering energy conservation and lifetime of the network while conceiving clusters. In the future work, we will study the different mobility model to evaluate our distributed hierarchical architecture.

Acknowledgment

This work is supported by the ANR 'Agence Nationale de la Recherche—France' within the project framework ARA/CLADIS.

References

1. Hu Y, Perrig A, Johnson DB. *Adriane*: a secure on-demand routing protocol for ad hoc network. In *Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom'02)*, 2002.
2. Johnson DB, Maltz DA. Dynamic source routing routing in ad hoc wireless networks. *Mobile Computing* 1996; **353**; 153–181.
3. Perrig A, Canetti R, Tygar JD, Song D. The TESLA broadcast authentication protocol. *Cryptobytes* 2002 **5**(2), 2002.
4. Sanzgiri K, Dahill B, LaFlamme D, Levine BN, Shields C, Belding-Royer EM. An authenticated routing protocol for secure ad hoc networks. In *Proceedings of Journal Selected Areas in Communication (JSAC)*, vol. 23, 2005; pp. 598–610.
5. Perkins CE, Royer EM. Ad-hoc on-demand distance vector routing. In *Proceedings of IEEE WMCSA'99*, 1999; 90–100.
6. Zhou L, Haas ZJ. Securing ad hoc networks. In *Proceedings of IEEE Network*, vol. 13, 1999; 24–30.
7. Zhou L, Schneider FB, van Renesse R. COCA: a secure distributed on-line certification authority. In *Proceedings of ACM Transactions on Computer Systems*, vol. 20, no. 4, 2002; 329–368.
8. Bechler M, Hof H-J, Kraft D, Pahlke F, Wolf L. A Cluster-based security architecture for ad hoc networks. In *Proceedings of IEEE INFOCOM'2004*, 2004; 2393–2403.
9. Zimmermann PR. *The Official PGP User's Guide*. MIT Press Cambridge: MA, USA, 1995.
10. Capkun S, Buttyan L, Hubaux J. Self-organized public-key management for mobile ad hoc networks. In *Proceedings of ACM International Workshop on Wireless Security, WiSe*, 2002; 52–64.
11. Chokhani S, Ford W, Sabett R, Merill C. Internet X.509 public key infrastructure certificate policy and certification practices framework. *Internet Request for Comments (RFC3647)*, November 2003.
12. Shamir A. How to share a secret. In *Proceedings of ACM Communications Journal*, vol. 22, 1979; 612–613.
13. Budakoglu C, Gulliver TA. Hierarchical key management for mobile ad-hoc networks. In *Proceedings of IEEE Vehicular Technology Conference (VTC'2004)*, vol. 4, 2004; 2735–2738.
14. Yi S, Kravets R. MOCA: mobile certificate authority for wireless ad-hoc networks. In *Proceedings of the 2^{nd} Annual PKI Research Workshop (PKI'03)*, 2003.
15. Dong Y, Go HW, Sui AF, Li VOK, Hui LCK, Yiu SM. Providing distributed certificate authority service in mobile ad hoc networks. In *Proceedings of Computer Communication journal*, vol. 30, 2007; 2442–2452.
16. Kong J, Zerfos P, Luo H, Lu S, Zhang L. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *Proceedings of International Conference in Network Protocols (ICNP'01)*, 2001.
17. Satizabal C, Hernandez-Serrano J, Forné J, Pegueroles J. Building a virtual hierarchy to simplify certification path discovery in mobile ad-hoc networks. In *Proceedings of Computer Communication journal*, vol. 30, 2007; 1498–1512, 2007.
18. Gerla M, Tsai JT-C. Multicluster, mobile, multimedia radio network. In *Proceedings of ACM/Baltzer Journal of Wireless Networks*. vol. 1, (no. 3), 1995; 255–265.
19. Chiang C, Wu H, Liu W, Gerla M. Routing in clustered multihop mobile wireless networks with fading channel. In *Proceedings of IEEE SICON'97*, 1997; 197–211.
20. Basu P, Khan N, Little T. A mobility based metric for clustering in mobile ad hoc networks. In *Proceedings of Distributed Computing Systems Workshop*, 2001; 43–51.
21. Inn I Er, Seah WKG. Mobility-based d-hop clustering algorithm for mobile ad hoc networks. In *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC'2004)*, 2004; 2359–2364.
22. Rachedi A, Benslimane A. A hiearchical distributed architecture to secure ad-hoc networks. *Research Technical Report,* LIA, 2006.
23. Kyasanur P, Vaidya N. Selfish MAC layer misbehavior in wireless networks. In *Proceedings of IEEE Transactions on Mobile Computing*, vol. 4, issue 5, 2005; 502–516.
24. Raya M, Hubaux J-P, Aad I. DOMINO: a system to detect greedy behavior in IEEE 802.11 hotspots. In *Proceedings of the 2nd international conference on Mobile systems, applications, and services (MobiSys'04)*, 2004; 84–97.
25. Guang L, Assi C, Benslimane A. Modeling and analysis of predictable random backoff in selfish environments. In *Proceedings of the 9-th ACM/IEEE international symposium on modeling, analysis and simulation of wireless and mobile systems (MSWiM'2006)*, Torremolinos, Malaga, Spain, 2006.
26. Rebahi Y, V. E Mujica-V, Sisalem D. A reputation-based trust mechanism for ad hoc networks. In *Proceedings of the Symposium on Communications (ISCC'05)*, 2005.
27. Rachedi A, Benslimane A. A secure architecture for mobile ad hoc networks. In *Proceedings of International Conference on Mobile Ad-Hoc and Sensor Networks (MSN'06)*, Lecture Notes in Computer Science, vol. 4325, Hong Kong, China, 2006; 424–435.
28. Sanchez M, Manzoni P, Haas ZJ. Determination of critical transmission range in ad-hoc networks. In *Proceedings of Multiaccess Mobility and Teletraffic for Wireless Communication*, 1999.
29. Krishnamachari B, Wicker SB, Bejar R, Pearlman M. Critical density threshold in distributed wireless networks. In *Communications, Information and Network Security*. Kluwer: Dordrecht, 2002; 1–15.
30. Berkeley UC, USC ISI. The network simulator ns-2. Part of the VINT project. Available from http://www.isi.edu/nsnam/ns, 1998.
31. Yi S, Kravets R. Quality of authentication in ad hoc networks. In *Proceedings of ACM, MobiCom2004*, 2004.

DOI: 10.1002/sec