A secure architecture for mobile ad hoc networks

Abderrezak Rachedi, Abderrahim Benslimane

A Secure Architecture for Mobile Ad Hoc Networks

Abderrezak Rachedi[1] and Abderrahim Benslimane[2]

[1] LIA/CERI, University of Avignon, Agroparc
BP 1228, 84911
Avignon, France
abderrezak.rachedi@univ-avignon.fr
[2] LARIM/Computer Engineering Department,Ecole Polytechnique of Montreal
P.O. Box 6079
Station Centre-ville Montreal H3C 3A7 Canada
abderrahim.benslimane@polymtl.ca

Abstract. In this paper, we propose a new architecture based on an efficient trust model and clustering algorithm in order to distribute a certification authority (CA) for ensuring the distribution of certificates in each cluster. We use the combination of fully self-organized security for trust model like PGP adapted to ad-hoc technology and the clustering algorithm which is based on the use of trust and mobility metric, in order to select the clusterhead and to establish PKI in each cluster for authentication and exchange of data. Furthermore, we present new approach Dynamic Demilitarized Zone (DDMZ) to protect CA in each cluster. The principle idea of DDMZ consists to select the dispensable nodes, also called registration authorities; 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 CA. With this approach, we can avoid the single point of failure in each cluster. This architecture can be easily extended to other hierarchical routing protocols. Simulation results confirm that our architecture is scalable and secure.

Keywords: wireless ad hoc networks, security, clustering algorithm

1 Introduction

In recent years, much interest has been involved in the design of Mobile Ad-hoc Network (MANET) technologies. Mobile ad-hoc networks are characterized by their self-configuration, open peer-to-peer network architecture, shared wireless medium, stringent resource constraints, and highly dynamic network topology. These characteristics make them vulnerable to security attacks. Existing security solutions for wired or wireless networks with infrastructure cannot be directly applied to MANETs. Designing security solutions for MANET is the nontrivial challenges. The goal of security solutions is to provide security services, such as

authentication, confidentiality, integrity, and availability to mobile users. In order to achieve this goal, we must develop some key management systems adapted to the characteristics of MANET.
In this article, we propose a new architecture based on an efficient trust model and distributed clustering algorithm for designing the specific public key management systems. Our trust model is based on PGP (Pretty Good Privacy) approach [1] adapted to MANET characteristics, like fully self-organized security proposed by Hubaux et al. in [2]. Our distributed clustering algorithm uses the trust level and mobility metric for the selection of the cluster head (CH) which becomes certification authority (CA) in the cluster. In order to secure the cluster formation, we propose a new scheme which uses dispensable confident nodes, called registration authorities (RA). It consists to provide dynamic demilitarized zone (DDMZ) at one-hope from the CA in each cluster. The role of RAs is to protect CA, by receiving requests of certification, filtering and treating these demands before forwarding them to the CA.

The rest of the paper is organized as follows. In section 2, we discuss the related work on current key management systems developed for MANET. In section 3, we describe our global architecture, our trust model and the distributed clustering algorithm. In section 4, we present the simulation results of our distributed hierarchical architecture. In section 5, we study and analyse the security of our system. The last section consists of the conclusion.

2 Related work

Several works have been proposed in the literature to deal with the security problems in ad hoc networks. Specially, we investigate the distributed CA approach using threshold cryptography scheme and a clustering concept.

2.1 Distributed CA approach using threshold cryptography

Having a single central authority to distribute the public key certificate to all nodes is not suitable for MANET, because this scheme is vulnerable to single point of failure like ARAN (Authenticated Routing in Ad-hoc Networks protocol) [13]. If this node is compromised, the entire network becomes compromised.

Zhou and Haas's idea consists on distributing the CA role among n nodes of the network using $(n, k+1)$ threshold cryptography scheme [11]. In this scheme the secret key is divided into n partial shares $(S1, S2, \cdots, Sn)$ where at least $k+1$ of n are partial shares which are needed to generate a secret S. The advantage is its increased availability, since any $k+1$ among n nodes in the local neighborhood of the requesting node can issue or renew a certificate. Another advantage is that any node, which does not have a secret share yet, can obtain a share from any group of at least $k+1$ nodes which has already a share. Unfortunately, this scheme has some drawbacks: First, the $k+1$ nodes must be initialized by a trusted authority. Second, the number k must be a trade-off between availability and robustness. Third, the system overloads the network because instead of

sending only one request for obtaining certificate or revocation the node must send at least $k + 1$ request (k traffic add in network).

2.2 A clustering concept

Mobile ad-hoc network may be represented as a set of clusters. Each cluster is represented by a cluster-head (CH) and gateway nodes which manage the communication with adjacent clusters. Among several secure solutions based on clustering ad hoc networks which we studied, one is a cluster based security architecture proposed by Becheler et al. [12]. This architecture use the threshold cryptography scheme (n, k) to distribute 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. After that, it must request at least (k) certificates from different CHs, whose association of these k certificates gives the network certificate. The drawbacks of the log-on procedure are as follow. 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 nodes, so that they can decide to guarantee or not). Second, even the guest has already W certificates from guarantors, it cannot succeed to have K certificates from CHs and it will not be certified. Third, the network traffic generated by each new node in these procedure is at least $2 * (W + K)$ packets. The Becheler's architecture has some other 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 dominate key among these different network's keys depends on the number of CHs of each network. The network which has maximum CHs will become dominant network and its network key remains private key of CA. These processes present a point of failure, because in this architecture any node can construct its own cluster. Thus, a set of malicious nodes can form their network with the maximum of CHs, and then attack the network in order to merger in the network and take the CA control.

This architecture does not contemplate the presence of two CHs in only one cluster due to the mobility of the nodes. Furthermore, the selection criteria of CHs are not mentioned in the paper. Also, to renew the network key, the intervention of a trust third party is needed so that it can subdivide the new key and distribute the fragment of the key over CHs. In the light of above factors, we believe that this architecture is not well adapted for Ad hoc environments.

3 Architecture

Firstly, we define a new trust model on which is based our architecture. 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 dynamic public key infrastructure with CA as clusterhead that will change according to topology changes. We propose a clustering algorithm based on this trust model. A new concept is proposed to protect CA node in each cluster based on dispensable nodes.

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

We assume that there are spare social relationships among nodes in order to establish trust relationship of any to-be-trust node with confident nodes. Also, every node 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 have initially k trust nodes these nodes have k-1 entries (known mutually) in their trust table.

3.2 Trust Model

In our trust model, we define trust metric (Tm) as continuous value on the [0..1] interval. A node i has a high trust value $(Tm(i) = 1)$, if it is known and exchanged keys over secure side channel (physical encounters and friends) with one or more of confident nodes [8][2]. Another manner to obtain a high trust value is that a node must prove its good faith by adapting a good behavior and a cooperation. If a new node is added in the trust table by one or more confident nodes all others confident nodes will be aware. This is because confident nodes update and exchange their trust tables. Each new unknown node starts with $Tm = 0.1$ its lower trust level.

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

- CA_k : Certification authority of cluster k which certificate public key of nodes belonging in the same cluster. CA_k has high trust level, Tm value must be equal one.
- $RA_{i,k}$: Registration Authority of cluster k assured by trust node i. The mean goal of RA is to protect CA against attackers that by DDMZ formation in order to prevent direct communication between unknown nodes and CA, for example, they treat and filter the requests of certification toward CA. Also, RAs must be confident nodes with $Tm(i) = 1$.

– $GW_{i,j}$: It is a gateway node ensuring a connection between two different clusters i and j. These nodes must be certified by two different CA. GW nodes must haves good trust level with $Tm(g) \in [0.7 - 1.0]$.
– $MN_{i,k}$: it represents a member node i belonging to the cluster k which success to pass from visitor to member status by well behaviour 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 $Tm(i) \in [0.5 - 0.7]$.
– $VN_{i,k}$: It is a visitor node i that belongs to cluster k, it has low trust certificate, because CA_k and $RA_{j,k}$ nodes need to have more information about node i behavior. Node i cannot communicate outside its cluster. It has a minimal trust level $Tm(i) \in [0.1 - 0.5]$.

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

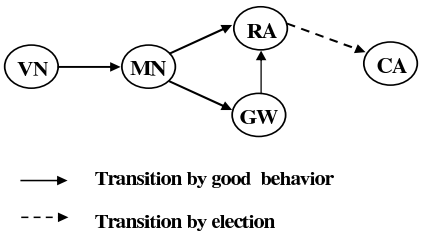

Fig. 1. State transition diagram

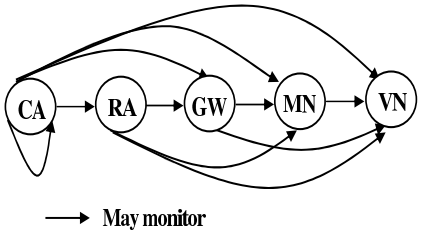

Fig. 2. Monitoring scheme

The hierarchical monitoring process consists to supervise behaviors of nodes. Each node with high trust value monitors its neighbors nodes with low trust value. Figure 2 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 {GW,MN,VN} status, also GW can supervise {MN,VN} status. Finally MN node can supervise only VN status but VN cannot supervise any node.

In our trust model, the trust relationship is ensured by CAs between clusters. A CA can recommend node, with certain trust level, belonging in its cluster to another CA. It is also ensured by RA in order to recommend nodes which belong in 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 to take the small trust value of nodes (eg. Trust between RA and GW is $min(1, w)$ where $w \in [0.7 - 0.9]$). Figure 3 shows two examples of certificate chain. The best trust chain (TC) is given in the case of b: $TC(b) > TC(a)$.

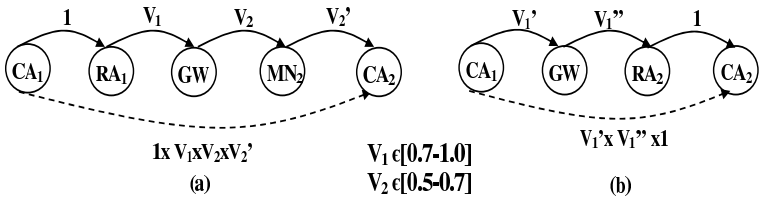

Fig. 3. Certificate Chain

3.3 Distributed Clustering Algorithm

A clustering network in our architecture is ensured by a Secure Distributed Clustering Algorithm (SDCA). The main rules of this algorithm are as follow:

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

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

The security parameter depends on 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 [10], max-connectivity algorithm [9]. In our algorithm, we adopt a mobility metric [3][7] because this strategy gives good result 33% of reduction in the number of cluster-head changes compared with last approach [10][9].

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 [4].

– 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.

$$(MAC_{K-}[CA, Hopcount, DTN, RM, ID - num])$$

This information permits to 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 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 (one-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 one hope or more depends on the value of d.

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 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.

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 [4].

Algorithm 1: Clustering Algorithm executed by confident node

```
When receiving a beacon by node j from node i;
begin
    Authentication do if (Tm(i)! = 1) then
    RejectBeacon(); Goto(end);
    else if (HopCount >= d) 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
```

4 Performance evaluation

We have implemented our clustering algorithm as described previously. We use Network Simulator (NS-2) [14] with CMU wireless extensions to simulate our algorithm. Simulation scenarios were generated with parameters listed in the table 1.

The movement of mobile nodes is randomly generated and continuous within the whole simulation periode.
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 4, 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 raison. 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 decreases and more of 150 m the network become more stable. However, while fixing d=2 we obtain less cluster-head compared to MOBIC and lowest-ID.

The cluster-head (CA) duration is the parameter which indicate the cluster stability. The longer CA duration means the system is more stable. The simulations with 100 nodes, $2500x2500$ of scenarios size and $12.5m/s$ of maximum speed gives 12.8 second of average CA duration.
Figure 5 shows the average number of different status of nodes in the network.

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
```

Table 1. Simulation parameters

Parameter	Value in our simulation
Number of nodes (N)	50
Network size (mxn)	$670x670m2$
Constant mobility	20 m/sec
Transmission Range	10 m - 125 m
Pause time	0.30 s
Broadcast interval (BI)	0.75-1.25 s
Discovery interval	10*BI s
Contention periode	3.0 s

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.

5 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,

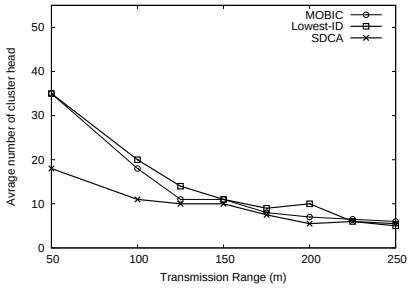

Fig. 4. Comaprison between different clustering algorithms

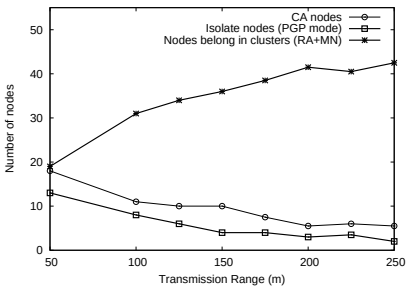

Fig. 5. Average number of different status of nodes

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.

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.
In order to evaluate the trust of CA authentication, we calculate the quality of authentication (QoA), so that, we apply attenuation factor to trust chain [6]. 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.

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

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 getaway nodes (GW).

The figure 6 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 two hop we note that, QoA decrease more fast with probability of malicious nodes than the case of one hop of cluster size.

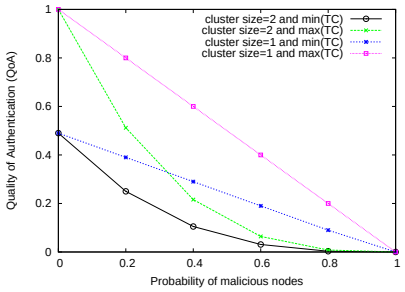

Fig. 6. QoA versus probability of malicious nodes

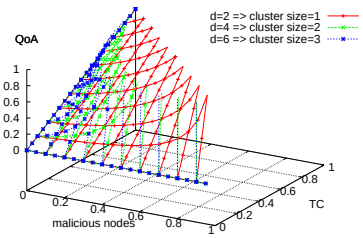

Fig. 7. QoA versus probability of malicious nodes and trust chain

The figure 7 shows the general case of QoA with different values of TC and probability of malicious nodes. We compare three case of cluster size 1, 2, and 3 hop, we remark the best value of QoA is in the case of one hop of cluster size, low value of malicious nodes and high TC.

According to the last figures 7 and 6, we can conclude that, more of the cluster size is large, the risk to have weak QoA is hight.

6 Conclusion

In this paper, we have proposed a new architecture based on our trust model and clustering algorithm in order to distribute a certification authority (CA).

Our clustering algorithm is based on two parameters: security and stability. The security factor is related to the trust model; only confident nodes can become cluster-head 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.

Our architecture ensures the security and availability of public key authentication in each cluster. This architecture is adapted to any topology changes.

Simulation results of our clustering algorithm showed the improvement of clusters stability compared to MOBIC and Lowest-ID algorithms. Furthermore, 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. Our future work is to study and analyse our architecture in order to evaluate the resistance degrees of DDMZ faced to different DoS attacks.

References

1. Philip R. Zimmermann: The official PGP user's guide.MIT Press Cambridge. "MA, USA. (1995)
2. S. Capkun and L. Buttyan and J. Hubaux: Self-Organized Public-Key Management for Mobile Ad Hoc Networks. ACM International Workshop on Wireless Security, WiSe. **2** (2002) 52–64
3. P. Basu and N. Khan and T. Little: A mobility based metric for clustering in mobile ad hoc networks. In Proceedings of Distributed Computing Systems Workshop. (2001) 43–51
4. A. Rachedi and A. Benslimane: A Hiearchical Distributed Architecture to Secure Ad-Hoc Networks. Research Technical Report LIA. (2006)
5. M. Gerla and J. T.-C. Tsai: SMulticluster, Mobile Multimedia Radio Networks. Wireless Networks. (1995) 255–256
6. S. Yi and R. Kravets: Quality of Authentication in Ad Hoc Networks. ACM, MobiCom2004. (2004)
7. I. Inn Er and Winston K.G. seah. Mobility-based d-hop Clustering Algorithm for Mobile Ad Hoc Networks. (2004)
8. S. Capkun and J. P. Hubaux and L. Buttyan: Mobility Helps Peer-to-Peer Security. IEEE Transactions on Mobile Computing. **5** (2006) 48–60
9. C. Chiang and H. Wu and W. Liu and M. Gerla: Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel. IEEE Proceedings of SICON'97. (1997) 197–211
10. M. Gerla and J. T.-C. Tsai: Multicluster, Mobile Multimedia Radio Networks. Wireless Networks. (1995) 255–256
11. Lidong Zhou and Zygmunt J. Haas: Securing Ad Hoc Networks. IEEE Network. **13** (1999) 24 –30
12. Marc Bechler and Hans-Joachim Hof and Daniel Kraft and Frank Pahlke and Lars Wolf: A Cluster-Based Security Architecture for Ad Hoc Networks. INFOCOM2004. (2004)
13. Kimaya Sanzgiri and Bridget Dahill and Daniel LaFlamme and Brian N. Levine and Clay Shields and Elizabeth M. Belding-Royer: An Authenticated Routing Protocol for Secure Ad Hoc Networks. Selected Areas in Communication (JSAC). **23** (2005) 598–610

14. UC Berkeley and USC ISI: The network simulator ns-2. Part of the VINT project. Available from http://www.isi.edu/nsnam/ns. (1998)