Towards the Definition and Performance Evaluation of a Trust-based Security Mechanism in a MANET Environment

Aida Ben Chehida Douss
Higher School of Communication, Sup'Com
University of Carthage
Tunis, Tunisia

Ryma Abassi
Higher School of Communication, Sup'Com
University of Carthage
Tunis, Tunisia

Sihem Guemara El Fatmi
Higher School of Communication, Sup'Com
University of Carthage
Tunis, Tunisia

***Abstract*—A MANET (Mobile Ad hoc NETwork) is a self-organized wireless network with mobile and collaborating nodes without any pre-established infrastructure. Because of these specifications, securing MANET constitutes a hard and challenging task that has attracted many researchers. For our concern, we proposed in a previous work a Mobility-based Clustering Algorithm (MCA) as well as a Trust management scheme for MCA (TMCA) to secure routing behaviors. MCA organizes nodes into clusters with one-hop members and elected Cluster-Heads (CHs), and allows the network maintenance in the presence of mobility. TMCA on the other hand locates malicious nodes and isolates them based on their reputations. The work presented in this paper tries (for network stability and performance improvement) to extend first the TMCA scheme with a delegation process, the whole proposition is baptized DTMCA, then, to evaluate the performances of the whole DTMCA scheme using simulation experiments. DTMCA scheme offers to a CH a new functionality: the delegation of its functions to one of its cluster member in case of displacement or energy depletion. DTMCA is based on two phases: initialization and notification. During initialization, a member node is elected whereas notification phase is used to inform nodes about the identity of the new CH. Some simulation experiments conducted to evaluate the performances of DTMCA scheme and presented at the end of this paper showed a significantly improvement in terms of throughput and lost packets ratio.**

Keywords*: *MANET, clustering, security, trust management, delegation.

I. INTRODUCTION

Mobile Ad hoc NETwork (MANET) is an autonomous system where wireless and battery powered mobile nodes cooperatively maintains network connectivity without central administration or established infrastructure [1]. Due to these characteristics, all networking functions must be performed by the nodes themselves. Having that each node in MANET has to act as both host and router, classical routing protocols cannot be used in such environment. Hence, some specific ones have been proposed. Unfortunately and due to MANET characteristics, malicious nodes can easily compromise the routing protocol functionality by disrupting the route discovery process and then corrupt network functioning and degrade its performances. Securing MANET has become then a prevalent research area over the last years.

In a previous work, we proposed a Trust management scheme for Mobility-based Clustering Algorithm (TMCA) [2] to detect and isolate malicious nodes. This scheme is built upon a new Mobility-based Clustering Approach (MCA) [3] to reduce network overhead and handles network topology dynamicity. Clustering in MANET is used to organize nodes into groups (clusters) characterized by cluster-head (CH) and member nodes [4]. MCA organizes nodes into clusters with one-hop members and elect CHs according to the highest weight calculated using two parameters: the residual energy and the mobility. This organization is also maintained in the presence of mobility. For security aims, we have proposed around MCA [3], a trust management scheme TMCA that detects malicious routing behavior based on CHs direct observations as well as alerts exchanged between them. Four modules constitute the TMCA scheme (1) a monitoring module to detect malicious nodes, (2) a reputation module to update reputation values, (3) an isolation module to discard malicious nodes and (4) an identity recognition module to assess alerts sources.

In order to improve network performance and to maintain its stability, we propose in this paper to extend the TMCA scheme with a Delegation process TMCA based (DTMCA). Delegation in fact is the process allowing a node to share or transfer its functionalities [5]. Using delegation; a node will be able to give its functionalities to another node when it is no longer able to perform them. For our concern, DTMCA uses the delegation process to allow the delegation of the CH functionalities to one of its cluster member (called *delegatee*) in case of displacement or energy depletion. Using DTMCA, the CH elects the most honest member node having the lowest weight. Two phases characterizes DTMCA: (1) the initialization phase choosing the *delegatee* member node and (2) the notification phase informing this *delegatee* and other cluster members about the identity of the new CH. DTMCA improves network performances and contributes also to the stability of clusters by avoiding the re-invocation of the clustering approach in case of CH failure.

The remaining part of this paper is structured as follows: Section 2 reviews some basics trust modeling concepts. Section 3 recalls a previously proposed Trust Management scheme called TMCA. This latter is based on a newly introduced mobility-based clustering algorithm MCA. Section 4 is concerned with the main contribution of this paper: a novel delegation process built over TMCA, baptized DTMCA. Section 5 presents some simulation results showing some DTMCA performances compared with a simple routing protocol. Finally, section 6 concludes this paper by summarizing its main contributions.

II. RELATED WORKS

Several works has been proposed in the context of using trust to secure MANET and to ameliorate its performances. Most of them are based on reputation in order to detect and isolate malicious nodes. Reputation is a perception a party creates through past actions about its intentions and norms [6].

Marti et *al.* [7] proposed a reputation-based scheme consisting of a Watchdog monitoring node behaviors and a Pathrater collecting reputation and reacting. Watchdog use observation-based techniques to detect misbehaving nodes and report observed misbehavior back to the source of the traffic. Pathrater manages trust and route selection based on these reports. This allows nodes to choose better paths along which to route their traffic by routing around the misbehaving nodes. However, the scheme does not punish malicious nodes; instead, they are relieved of their forwarding burden.

CONFIDANT was proposed by Buchegger and Boudec [8]. This approach has four main components: a monitor, a reputation system, a path manager and a trust manager. It is used to detect and isolate misbehaving nodes by combining monitored and experienced information of a node's behavior with warnings reported from other nodes. CONFIDANT implements a punishment-based scheme by not forwarding malicious nodes' packets. The major drawback of this approach is that it uses only negatives experiences and is vulnerable to false positive detection due mainly to network congestion. Since this protocol allows nodes in the network to send warning to each other, it could give more opportunities for attackers to send false alarm messages.

Michiardi and Molva [9] proposed CORE, a COllaborative REputation mechanism based on Watchdog. CORE uses a reputation mechanism differentiating between subjective reputation (observations), indirect reputation (positive reports by others), and functional reputation (task-specific behavior).These latter are weighted in order to obtain a combined reputation used to make decisions about cooperation or gradual isolation of a node. A main characteristic of this mechanism is that only positive reputation information is exchanged. However, this may limit its reliance on positive reports without the facility to submit negative feedback.

In a recent work [10], Abassi et *al.* proposed to deal with delegation in a trust based MANET. In this work, authors proposed a modeling for delegation management during its initialization, negotiation and revocation based on trust relations. More precisely, they proposed to initialize delegation based on reputations evaluation, negotiation is achieved through *trustor's* request whereas the third activity concerns revocation of delegations and consequently all its associated actions.

To our best knowledge, there is no existing work benefiting from clustering, reputation concepts and delegation to secure routing process in MANET, to ameliorate its performances and to maintain its stability. The main proposition of this paper is then a delegation process DTMCA based on an already proposed trust management scheme TMCA and on a mobility-based clustering algorithm MCA. The performances of the whole proposed DTMCA scheme are also evaluated using a simulation experiments.

III. TMCA: TRUST MANAGEMENT SCHEME MCA-BASED

Recently, we proposed TMCA, a trust management scheme based on the proposed MCA approach in order to build a secured MANET environment [2]. TMCA scheme detects and isolates malicious routing behaviors based on CHs direct observations as well as alerts exchanged between nodes.

In this section we recall first the TMCA basic concepts, and then we present its modules.

A. MCA: the Mobility-based Clustering Algorithm

MCA is based on the following assumptions.

- The network will be organized into clusters with one-hop members and a CH.
- Each member node should belong to one cluster.
- The election of the CH is made based on a combined weight calculated according to two parameters: The residual energy and the mobility. The elected CH should have the smallest weight.
- MCA adapts clusters following network topology changes i.e. node addition, displacement or failure.

Two phases are then proposed: setting up and maintenance. The setting up phase is based on: (1) the cluster identification and (2) the CH election. The deployment of these two components assumes that each node in the network has already performed a preprocessing phase. This latter is used to discover its neighborhood through the HELLO and ACK_HELLO messages, to compute its weight and to broadcast it using the WEIGHT message.

After the preprocessing phase, the cluster identification component is performed. This component is used to generate the restricted (one-hop) neighborhood noted *RN* where each node *i* generates its RN_i: two nodes *j* and *k* belong to the same RN_i, if *j* and *k* are neighbors. If node *i* has more than one *RN*, the chosen *RN* will be the one having the least mobility. This *RN* represents then the node's cluster. Cluster identification component is completed by broadcasting the *RN* through the RN message. The second component in the setting up phase is the CH election. The node having the smallest weight among its *RN* neighbor weights declare itself as CH using the CH message. All nodes belonging to the same cluster as this CH join it as members by broadcasting a JOIN message. Let us note that used messages are detailed in Table I.

The main contribution of MCA concerns mobility handling in clustering environment. This is done during the maintenance phase. Two topology changes have been handled in this phase: (1) the failure of a node and (2) the displacement or arrival of a node.

Link failure handling. When a node *i* detects the failure of its one-hop neighbor *j*, three cases are conceivable: (1) node *i* is a CH and node *j* its cluster member, in this case, node *j* is simply dropped from the *i*'s cluster and neighbor table, (2) node *i* is a CH and node *j* is not its cluster member, CH *i* drops node *j* from its neighbor table and (3) node *i* is a member node and node *j* its CH, in this case, if node *i* has the lowest weight among its *RN* neighbors, it declares itself as CH, else it waits for a CH message from another node.

New link handling. When a node *i* detects a new coming or a moving node *j* in its neighborhood, two cases are conceivable. (1) node *i* is a CH, it checks if *j* is neighbor with all its cluster members. If it is the case, CH *i* adds *j* into its cluster, else CH *i* creates a new cluster with node *j* and delegates its functionalities to one of its cluster member. The delegation concept is explained in the DTMCA section. (2) node *i* is a member node, it checks if node *j* is not neighbor with its CH *k*, if it is the case, node *i* notifies to its CH *k* the existence of node *j* and waits for its CH decision. If *k* authorizes such action, *i* sends to *j* the New_CH message with a flag set to 2 and a cluster is created containing nodes *i* and *j*.

B. *TMCA basic properties*

The proposed TMCA scheme is based on the following properties:

- Each CH in the network uses the Watchdog mechanism through the promiscuous mode to monitor the behavior of its cluster members.
- Each CH maintains a reputation table associating each node with its reputation value. In fact, we propose reputation values ranging from -3 to +3 with discrete values such that:
  - *If* rep ∈ [-3, 0[→ Malicious node.
  - *If* rep == 0 → Neutral node.
  - *If* rep ∈]0, +3] → Innocent node.
- Once elected, each CH is associated with the reputation value +3.
- New arriving nodes are associated with the neutral reputation value 0.
- Each CH updates the reputation value of its member nodes according to detected events.
- TMCA scheme is based on four modules: (1) the monitoring module detecting member behaviors, (2) the reputation module updating member's reputation, (3) the identity recognition module assessing alerts sources and (4) the isolation module isolating misbehaving nodes.
- TMCA modules compose all nodes but are actives only for CHs.
- A rehabilitation mechanism is also used to rehabilitate node having well behaved for a given period of time.

C. *TMCA scheme description*

Fig. 1. depicts TMCA scheme and modules within each CH in the network. Each elected CH monitors the behavior of its cluster members using the monitoring module. This module is based on the Watchdog mechanism. The CH may detect two kinds of events: (1) a positive event i.e. the member node forwards the packet and do not modifies it, (2) a negative event i.e. the member node don't forward the packet or modifies it. As soon as a positive or a negative event is detected, the reputation module is triggered to update the reputation value of the corresponding member node. However, if a positive event is received, the reputation module increments the reputation value of the member node by +0.2. However, if a negative event is detected, the reputation value is decremented by -1 if the event is "Packet dropping" and by -2 if the event is "Packet modification". In fact, the Watchdog mechanism may be faked by collision a problem [7] that is why we choose to punish less severely the dropping packet event.

When the reputation value of a member node falls below a minimum value -3, the isolation module is triggered to isolate the member node and to inform other CHs using an ALERT message. The malicious node is also added into a blacklist and all routes containing this node in the routing table are deleted.

Let us note that monitoring modules belonging to different CH can communicate through an ALERT message. This latter contains a notification about a malicious node, a reputation value of a detached member node or a rehabilitated node. Once received, the ALERT message is passed to the identity recognition module where the source of the message is checked i.e. whether it is a CH or not.

TABLE I. EXCHANGED MESSAGES AND NOTATIONS

Message	Meaning
HELLO (my_ID, my_M)	Notifies neighbors about my ID and my relative mobility M.
ACK_HELLO (my_ID, list_my_neighbors)	Notifies neighbors about my ID and my one-hop neighbors.
WEIGHT (my_ID, my_W)	Notifies neighbors about my ID and weight.
RN (my_ID, my_RN)	Notifies neighbors about my ID and RN.
CH (CH_ID, CH_Member)	Notifies RN neighbors about my role: I am a CH, my ID is CH_ID and my members are CH_Member.
JOIN (my_ID, CH_ID)	Notifies neighbors that I am going to join the cluster whose CH's ID is CH_ID.

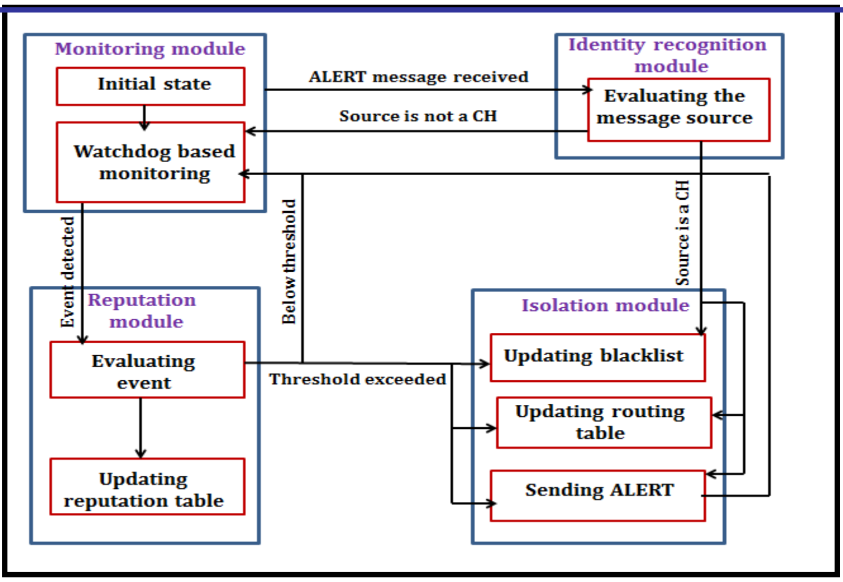

Fig.1. Proposed TMCA scheme and modules within each CH.

TMCA scheme proposes also a rehabilitation mechanism to rehabilitate node having well behaved for a given period of time. When a malicious node behaves well, its reputation value is incremented by 0.1. As soon as its reputation reaches the neutral value 0, the node is deleted from the blacklist and the CH informs other nodes in the network about the rehabilitated node.

IV. DTMCA: A DELEGATION PROCESS TMCA-BASED

In this section, we try to improve network performances and to maintain the stability of clusters (by avoiding the re-invocation of the clustering approach in case of CH failure) by extending the proposed trust management scheme TMCA with a delegation process DTMCA. Let us recall that delegation is the process whereby a node can share or transfer its functionalities. For our concern, DTMCA uses the delegation process to allow the delegation of the CHs functionalities to one of its members when it is not able to perform them.

A. DTMCA basic properties

DTMCA is based on the following properties:

- DTMCA process is triggered when the residual energy of the CH reaches a minimum threshold or during the clustering maintenance phase i.e. when the CH is obliged to create a new cluster with a new coming node as explained in Section III.A.
- Each CH can delegate its functionalities to one of its members through an election process.
- To ensure security, the chosen member node should have the highest reputation value, be stable and with enough residual energy.
- DTMCA is built upon two phases (1) the initialization phase to choose the *delegatee* member node and (2) the notification phase to inform the *delegatee* and other cluster members about the identity of the new CH.
- DTMCA improves network performances by avoiding the re-invocation of the setting-up clustering algorithm phase when the CH is not able to perform its functions.

In the following, these two steps are detailled.

B. DTMCA: The initialization phase

Initialization is the first DTMCA process phase. It is triggered when the CH have to delegate its functionnalities to one of its cluster member. In this case, the CH selects member nodes having the highest reputation value in the reputation table. Then, it selects the members with the lowest weight among these selected nodes. When more than one cluster member is selected, the chosen one is the node having the highest identifier. This assumption was made in order to avoid a blockage situation during the CH election.

The elected member node is then the node having the highest reputation value and the lowest weight value and consequently the most honest and stable member node as well as the one having the highest energy value.

Algorithm 1 depicts the initialization procedure using the following notations:

- *i*, the current node executing the procedure.
- Cluster$_i$, the set of nodes in *i*’s cluster.
- rp_value$_i$, *i*’s reputation value.
- ID_value$_i$: *i*’s identifer.
- W$_i$: i’s weight value.
- nb_table$_i$: *i* ’s neighbor table.
- rp_table$_i$: *i*’s reputation table.
- Send$_j$ Msg (), node *i* sends to node *j* a message Msg.
- Send*Msg(), node *i* broadcasts a message Msg to all its one-hop neighbors.

```
DTMCA_Initialization Procedure
Begin
        IN (rp_table$_i$)
        SELECT V ∈ Cluster$_i$ / rp_value$_V$ == MAX (rp_value)
                FROM SELECTED V:
                IN (nb_table$_i$)
                SELECT D ∈ V /
                W$_D$ == MIN (W)
        If (D> 1) Then
                Begin
                SELECT d ∈ D /
                ID_value$_d$ == MAX (ID_value)
                End
End
```

Algorithm 1. DTMCA initialization procedure

C. *DTMCA: The notification phase*

Once the initialization phase is performed, the CH unicasts a delegation request through the Del_REQ message to the *delegatee* node including its identifier as well as the *delegatee*'s identifier. When the member node receives this message and accepts the delegation request, it replies with the Del_REP message including its identifier.

Upon receiving this message, the CH shares its reputation table with the new CH and notifies its cluster members as well as other CHs in the network about the identity of the new CH using the Del_NOTIF message. This message includes old and new CHs identities. These messages are defined in Table II.

Algorithm 2 depicts the notification procedure.

```
DTMCA_Notification procedure
Begin
        Send$_d$ (Del_REQ (i, d))
        If (Receive$_d$ Del_REP (d)) Then
                Begin
                Share (rp_table$_i$)
                Send$_{Cluster_i}$ (Del_NOTIF (i, d))
                End
        Else Wait (del_Timer)
                If (del_Timer is expired) Then
                Begin
                rp_value$_d$ := -3
                blacklist$_i$= balcklist$_i$ / {d}
                Send$_{Cluster_i,CHs}$ (ALERT (i, 0, d))
                rt_table$_i$:= rtable$_i$/ {d}
                End
End
```

Algorithm 2. DTMCA notification procedure

Let us note that if the CH does not receive a Del_REP message from the chosen member node during a fixed time Del_timer, the following actions are triggered:

- Setting the member's reputation value to -3 and blacklisting it.
- Informing cluster members and other CHs about this non cooperative node and considers it as malicious.
- Deleting all paths including the malicious node from the routing table.
- Performing the initialization phase to select a new member.

Once the notification phase is performed, each CH receiving the ALERT message (notifying about the new chosen CH *d),* adds the node *d* into its CH_list.

TABLE II. EXCHANGED MESSAGES AND NOTATIONS

Message	**Meaning**
Del_REQ (CH_ID, new_CH_ID)	Notifies the selected member node that it was choosen to be the new CH.
Del_REP (new_CH_ID)	Notifies CH that the selected member node agrees to be CH.
Del_NOTIF(CH_ID, new_CH_ID)	Notifies cluster member about the new choosen CH identity.

V. SIMULATION AND RESULTS

The aim of the following section is to study the performance of the DTMCA proposition by a simulation work. The simulation parameters used are listed in Table III.

TABLE III. SIMULATION PARAMETERS

Parameter	**Value**
Simulator	NS vers 2.35
Nodes number	10-45
Network size	1000m*1000m
Transmission range	250m

Data traffic	CBR (Constant Bit Rate):
	data payload=512 bytes
	Rate= 4 packets/s

Node bandwidth	2 Mbps
Mobility	Random-Waypoint Model 500 m/s with a pause time= 30s
Routing Protocol	AODV
Simulation time	100 sec

To measure the performance of DTMCA, we consider the following two performance parameters: the throughput and the lost packets ratio. The throughput measures the average rate of successful packet delivered over a communication channel. the lost packets ratio corresponds to the percentage of lost packets versus sent packets. Let us note that the comparison is made with AODV after the clustering achievement.

Figure 2 depicts the evolution of throughput over time for both AODV and AODV extended with DTMCA. It shows that throughput is very close in both implementations until the 30th second. But after this time, the throughput is improved for our implementation because with less percentage of malicious nodes, the average rate of successful packet delivery is increased.

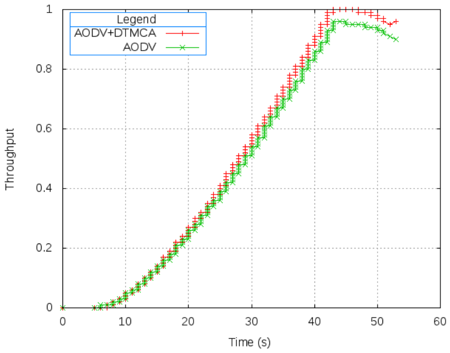

Figure 1. Throughput evolution with variying time

Figure 3 shows the lost packets ratio. We can observe that AODV and our implementation AODV+DTMCA have sensibly the same performancesfor the first 1000 packets. However, for the next 1500 packets, our simulation presents a lesser ratio of lost packets.This is due to the fact that malicious nodes are detected and isolated from the network.

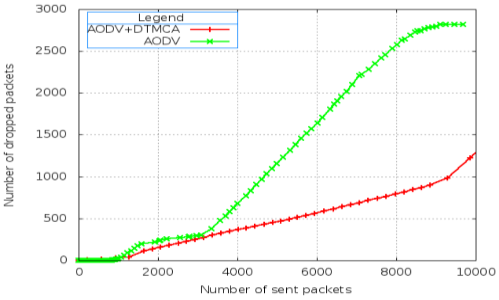

Figure 2. Lost packets versus sent packets

Let us note that we used only a reactive algorithm for this simulation but that our proposal can be applied to a proactive algorithm, too. In fact, the routing process should be secured in case of on-demand or table drive protocols. Even ifthese latters maintain routes for all destinations in the network, the establishement of these routes should be secured.

VI. CONCLUSION

The main contribution of this paper is a delegation process for MANET security based on clustering and trust management. Each clusteris composed bya set of one-hop members and an elected CH. The organization of the clusters is maintained following the mobility nodes. A trust management scheme accompanies this organization in order to secure routing by detecting malicious nodes and isolating them. In the proposed scheme, CH monitors the behavior of its cluster members and updates their reputation values following made observations.When the reputation value of a member node falls below a given thershold, it is considered as malicious and the CH informs its cluster members and the others CHs of the network and isolates it. In order to network performances and to maintain its stability, a delegation process DTMCA, extends the whole scheme. Using DTMCA, a CH will be able to give its functionnalities to a chosen member node when it is no longer able to perform them i.e. case of energy depletion or displacement.However, given its importance and criticality, delegation was associated to a security process. Thus, the selected member node should be honest, stable and with sufficient energy. DTCMA is then based on two phases: the initialization phase selecting a member node and the notification phase notifying the chosen member node and other cluster members with the identity of the new CH. Simulation results showed the efficiency of the proposed scheme in terms of throughput and lost packets ratio.

In future works, we expect enhancing our scheme with an access control process based on the proposed DTMCA scheme.

REFERENCES

[1] S. Kaushik and M. Kaushik, “Analysis of MANET Security, Architecture and Assessment”. International Journal of Electronics and Computer Science Engineering (IJECSE, ISSN: 2277-1956), 2012, vol. 1, no. 02, p.p 787-793.

[2] A. Ben Chehida, R. Abassi and S. Guemara El Fatmi, “A Reputation-based Clustering Mechanism for MANET Routing Security”. In Proceedings of the 8th International Conference on Availability, Reliability and Security ARES. September 2013, Reguensburg, Germany.

[3] A. Ben Chehida, R. Abassi and S. Guemara El Fatmi, “Towards the definition of a mobility-based clustering environment for MANET”. In Proceedings of the ninth International Conference on Wireless and Mobile Communications ICWMC. August 2013, Nice, France.

[4] A. Nassuora and A. Hussein, “CBPMD: A New Weighted Distributed Clustering Algorithm for Mobile Ad hoc Networks (MANETs)”. American Journal of Scientific Research ISSN, 1450-223X, Issue 22, 2011, pp. 43-56.

[5] M. B.Ghrobel-Talbi, F. Cuppens, N. Cuppens-Boulahia and A. Bouhoula, “Managing Delegation in Access Control Models”. In Proceedings of the 15th International Conference on Advanced Computing & Communication (AD’COM 2007), Guwahati, India, pp. 744-751, 2007.

[6] S. Ruohomaa and L. Kutvonen, “Trust Management Survey”. In Proceedings of iTrust 2005, LNCS 3477, pp. 77–92, 2005.

[7] S. Marti, T. Giuli, K. Lai, and M. Baker, “Mitigating Routing Misbehavior in Mobile Ad Hoc Networks,” In Proceedings of the Sixth Ann. Int’l Conference. Mobile Computing and Networking (MobiCom), pp.255-265, 2000, Boston MA, USA.

[8] S. Buchegger and J.Y.LeBoudec. “Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes Fairness In Dynamic Ad-hoc NeTworks”. In Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing Conference (MobiHOC), , June 2002, Lausanne.

[9] P. Michiardi, R.Molva, “Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks”. In Proceeding of IFIP-Communicatin and Multimedia Securtiy Conference August 2002,Slovenie.

[10] R. Abassi and S. Guemara El Fatmi, “Dealing with Delegation in a Trust-based MANET”. In Proceedings of the 20th International Conference on Telecommunication, ICT 2013, Casablanca, Morocco.