Mobile Netw Appl (2016) 21:149–160
DOI 10.1007/s11036-016-0703-y

# Autonomous Ant-based Public Key Authentication Mechanism for Mobile Ad-hoc Networks

**Parisa Memarmoshrefi**[1] · **Roman Seibel**[1] · **Dieter Hogrefe**[1]

Published online: 29 February 2016

**Abstract** In mobile ad-hoc networks (MANETs), where there is no centralized authority to provide authentication, trust and reputation mechanisms are applied to maintain security by identifying trustworthy and untrustworthy nodes. However, traditional authentication mechanisms are not viable for MANETs due to the lack of infrastructure and frequent topology changes. In this paper, we propose a self-organized and localized public key authentication mechanism based on ant colony systems. Every node generates its own public-private key pair, issues certificates to neighboring nodes and provides on-demand authentication services by means of gathering certificate chains towards a target node. Pheromone concentration left by ants along the path of the certificate chains represents the trust level of a node towards other nodes. This model is able to authenticate public keys by selecting the most trustworthy certificate chains gathered by ants and can identify and exclude certificate chains with malicious nodes.

**Keywords** Public key authentication · Security threat in trust and reputation systems · Ant colony optimization · MANETs

✉ Parisa Memarmoshrefi
memarmoshrefi@informatik.uni-goettingen.de

Roman Seibel
seibel@cs.uni-goettingen.de

Dieter Hogrefe
hogrefe@cs.uni-goettingen.de

[1] Institute of Computer Science, Telematics Group, University of Göttingen, Goldschmidtstraße 7, 37077 Göttingen, Germany

## 1 Introduction

Mobile ad-hoc networks are multi-hop wireless networks without infrastructure which are used in different civilian, military and emergency applications. In environments where cooperation to transmit information is unavoidable, because no trusted infrastructure is available, establishing secure communication is an essential issue. A possible scenario could be the following: in a post-disaster infrastructure-less area, operation of search and rescue can necessitate encrypted exchange of information between sender and receiver by using neighboring nodes, which forward the message towards the receiver. Hence in order to encrypt the message, the sender needs a public key of the receiver, which he can use to encrypt the message, so that only the receiver can decrypt it with his private key. In an infrastructure environment, the receiver's public key would be issued and certified by a trusted third party, such as a network operator. In an ad hoc network the receiver's public key must be created without the trustworthiness of the network operator and verified by the sender in a distributed fashion. Since authentication is the most important and basic part of any secure communication, in this work we consider the authenticity of a node as the context of trust in the authentication process. In order to provide secure network communication a key distribution procedure over basically insecure channels is necessary. A set of trust relationships is required to be built prior to authentication in this key distribution procedure.

A classification of authentication mechanisms in MANETs is presented in [1], identifying three different key management schemes: 1-central certification authority (CA) systems, which are not suitable for dynamic environments; 2-distributed CA systems, where $n$ nodes in a MANET collectively perform the task of a CA; 3- self CA systems which are based on a web of trust. Latter model allows nodes to become an individual CA, generate their own keying material and issue public key certificates for themselves and for others based on their trust values towards neighbors. A node's identity, here node *s*, is proven by node *b* as in the following procedure: Node *b*, obtains the public key and identity of node *s*. Node *b,* if it trusts *s,* composes a certificate that lists *s'* identity and public key. Node *b* signs the certificate with its own private key, verifying that *s'* public key correctly belongs to *s*. Other nodes can therefore obtain this certificate, issued by *b* towards *s* public key. In case these 3rd party nodes have the public key of *b*, they can decrypt the certificate and verify, that *b* "speaks" for *s* correct binding of public key and identity. Each node maintains a local certificate repository, and performs the public key authentication via a chain of certificates.

However there still exist security threats in this trust model. A number of security threats are presented in [2] which in general are relevant to trust and reputation systems. Therefore one of the most important subjects coping with dishonest misbehaving nodes along the certificate chains, that try to make other nodes trust in false identity-to-public-key bindings.

To mitigate the problem, we propose an on-demand, trust-based public key management system based on an ant colony communication principle [3]. The dynamic nature of ad hoc networks, caused by the mobility and the changing behavior of nodes, makes ant colony optimization an appropriate choice for the distribution of trust values. In our proposed scheme each node creates its own public-private key pair, issues certificates to neighboring nodes and stores the trust level of nodes in its repository. Reactively a node performs public key authentication by sending out ants towards the target node. The task of ants is to find the most trustworthy certificate chain. Through building a certificate chain, the ants, when successfully found a path to the destination node on their way back, leave traces of pheromone along the path representing a path trust trace. Despite of misbehaving nodes each node can make a suitable decision about the validity of the public key of a target node.

The rest of the paper is organized as follows: Section 2 represents related work. Section 3 includes the trust model and security threats relevant to our model. The ant colony system is described in section 4 and a description of our proposed model is presented in section 5. Some experimental results are presented in section 6 and finally section 7 provides the conclusion.

## 2 Related work

A public key certificate is a data structure in which a public key is bound to an identity and signed by the issuer of the certificate. In PGP [4] certificates are mainly stored in a centralized certificate repositories. [5] proposes a self-organized public key management where certificates are stored and exchanged by the nodes. The main problem of this scheme is its large overhead for storing the approximate global certificate graph. To solve this problem, authors in [6] propose an on-demand public key management solution. In this scheme all certificates need to be issued and trusted locally. A certificate chain can be obtained hop-by-hop, as long as a route is discovered between source node and destination node. [7] and [8] propose a solution based on an existing web of trust in integration with the AODV routing protocol.

So far several trust and reputation systems have been proposed, for dealing with malicious behaviour in different domains such as human social networks, e-commerce [9], peer-to-peer networks [10–12], mobile ad-hoc networks CONFIDANT [13] and CORE [14] and sensor networks [15, 16].

Each Trust model is composed of two main components: trust computation model and trust evidence distribution system. The distribution part is the basis of the computation part. TACS [12], AntRep [17] and [18] proposed trust-reputation evidence distribution using an algorithm based on bio-inspired ant colony systems in order to provide guarantees of network resource availability and trustworthiness. They are a reactive evidence distribution scheme and ants are sent out only when a request is made. In ant colony systems the main principle behind the interaction is called stigmergy, which means that the trace left in the environment by an action encourages the performance of the next action, by the same or different agent (ant).

Our work is similar to AntRep [17] and [18], we apply the ant colony optimization for trust evidence distribution, but there are important differences between our work and the state of the art. In [17] the authors assume that each peer provides a unique peer identifier, which is not an assumption in our design. In [18] evidence is in form of a trust certificate that contains a public key which is authorized by a private key of a signer. The authors have a strong assumption that there is a limited number of signers in the network, that their public keys are well known and authenticated and their private key cannot be compromised; However in our work we do not have such trusted signers. Every node in the network could play a role of a certificate issuer. As there are no predefined trusted certificate authorities, the system itself should be able to discriminate the trusted signer from malicious signers who aim at devastating the authentication process.

Our proposed incentive mechanism is adaptive to the environment with malicious public key signers. It evaluates the certificate chains gathered via a request that source nodes need to find the public key of a destination. Our certificate chain evaluation process does not deal with malicious node detection, instead it identifies a chain consisting of malicious nodes. Our mechanism is able to retrieve the public key certificate of a destination by excluding malicious signers from the public key authentication process by means of: Rewarding nodes within a trusted certificate chain and punishing of nodes within an untrusted chain, by updating pheromone levels.

## 3 Trust model and security threats

The trust model of our scheme is based on a web of trust of public key certificates that guarantees the bindings of the public keys to their related user identities. As an example $\text{Cert}_{i\to j}$ denotes the certificate that i issued for j and signed with its private key, $\text{Sig}_i$, to show the binding of node j's identity, $\text{ID}_j$, and its corresponding public key $\text{PK}_j$. In addition certificate $\text{Cert}_{i\to j}$ should contain the identity (network address) of issuer, $\text{ID}_i$, certificate validity time, T, and trust value or confidentiality, $t_{ij}$, representing the level of issuer's assurance of the binding of $\text{ID}_j$ to its corresponding public key, $\text{PK}_j$. We consider this web of trust a certificate graph G(V, E), whose set of vertices, V, represents the public keys and the set of edges, E, represents the certificates.

Each node periodically, depending on the expiration time of certificates, creates direct edges to its neighbors and issues certificates to them, if there is an acceptable confidentiality level for binding neighbors' ID to their corresponding PK. When a node, S, wants to authenticate the public key of another node, D, which is not located in radio range of S, a chain of valid certificates from S to D is required, Fig. 1.

In our example the certificate chain from S to D is $\{Cert_{S\to B}, Cert_{B\to C}, Cert_{C\to D}\}$. Every certificate in the chain will be verified with the public key of the previous certificate in the chain. But how to verify the certificate chain and how to choose the certificate chain composed of trustworthy nodes is still a problem that we discuss in the following parts.

### 3.1 Trust metrics

Trust metrics is a measure that represents the assurance that a requesting node can obtain the public key of the destination node correctly, through the certificate chain. In this chain fashion, trust transitivity plays a great role which is based on recommendations between entities. However, there is a difference between trusting an entity to provide a specific service and trusting an entity that recommends someone who can provide the service [19]. Trust in the service object is functional trust, while trust in recommending agents is referral trust. In our model we consider the functional trust as the honest binding rate, i.e. the number of correct binding signs over all trials. On the other hand referral trust is the dissemination of these scores to the relying nodes that can be considered as recommendations.

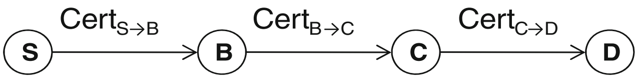

**Fig. 1** Certificate chain

Any node in the network can calculate the trust value of another node's public key if there is a physical communication and consequently a certificate chain between the two nodes using formula 1.

$$t_{SD} = \prod_{k=1}^{n} t_k \tag{1}$$

$t_k$ is the trust value between two directly connected nodes on the certificate chain from node S to node D. n is the number of hops between the source and the destination. It is obvious that the trust in another's public key fades along the path of recommendation.

### 3.2 Security threats

There is no guarantee in such decentralized public key management systems that all nodes act correctly and honestly. In general two types of functional and referral misbehavior threatens the security of our trust-based system.

Functional misbehavior occurs when a node or a group of nodes refuses to act correctly in service provision. In our authentication model it takes place by not participating in the authentication process or issuing a false certificate with an incorrect binding of a key to an identity. Impersonating another node is an example of functional misbehavior. A malicious node B may issue a certificate that binds another node's identity, $\text{ID}_C$, to its public key, $\text{PK}_B$, and signs it with its private key $\text{Pr}_B$, in the signature form of SigB (Fig. 2). The aim of the malicious node is eavesdropping a messages sent to j. Another example is binding the public key of node k, $\text{PK}_k$, to $\text{ID}_j$; although it should be bound to $\text{ID}_k$.

In the second type of misbehavior, referral misbehavior, a malicious node tries to trick other nodes by providing dishonest recommendations by manipulating the confidence in the authenticity of a given key.A well-known example of this kind of misbehaving is a Sybil attacker [20]; where the malicious node creates multiple identities for its single entity and aims at pretending to be separate individual nodes in the network.

As the false certificate is signed by many Sybil nodes, it could be considered as a correct certificate to non-Sybil nodes (Fig. 3).

The aim of our proposed model is a self-organized authentication mechanism which enables a defense mechanism against these two types of misbehavior. In this paper we concentrate on misbehaving nodes that try to

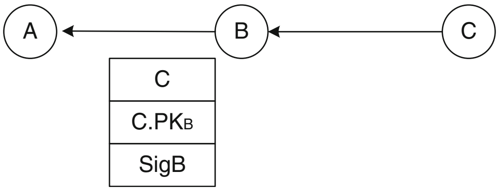

**Fig. 2** Node B impersonates node C

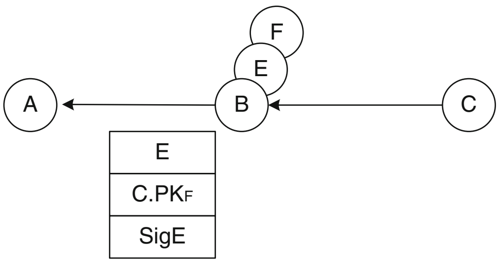

**Fig. 3** Sybil node B generates multiple IDs and corresponding key pairs in order to issue and sign faked and dishonest certificates

defect the authentication service by disseminating false information and provide a fake certificates for public keys of destinations.

## 4 Ant colony optimization

In a "trusted" network based on ant colony optimization, mobile agents, called artificial ants spread through the network from source to destination in order to find the most trustworthy path towards a destination node. The ants remember the visited nodes they passed, and deposit 'pheromone' on them. Ants are attracted to paths with higher pheromone concentration. When an ant wants to move from a starting node S toward a destination it chooses one of the neighboring nodes of S, i, with the probability defined by following transition rule:

$$p(\mathrm{S},\mathrm{i}) = \frac{[\tau_{\mathrm{Si}}]^{\alpha}.[\eta_{\mathrm{Si}}]^{\beta}}{\sum_{\mathrm{j}\in\mathrm{N(S)}}[\tau_{\mathrm{Sj}}]^{\alpha}.[\eta_{\mathrm{Si}}]^{\beta}}\ ;\qquad \sum_{\mathrm{i}\in\mathrm{N(S)}} p(\mathrm{S},\mathrm{i}) \ = 1 \tag{2}$$

where $\tau_{\mathrm{Si}}$ is the pheromone deposit on the edge between S and i, $\eta_{\mathrm{Si}}$ is the goodness value of the link between S and its neighbor node, N (S) is the list of neighboring nodes of S and $\alpha$ and $\beta$ are the weights for balancing the deposited pheromone and the goodness value of the edge respectively.

The following transition rule is used to provide a pseudo-aleatory path choice:

$$r = \begin{cases} argmax\ \ \mathrm{j}\in\mathrm{N(S)}\,[\tau_{\mathrm{Sj}}]^{\alpha}.[\eta_{\mathrm{Si}}]^{\beta} & \text{if}\quad \mathrm{q}\le\mathrm{q}_0 \\ R & \text{otherwise} \end{cases} \tag{3}$$

where $r$ is the next chosen node by an ant in its next movement, $\mathrm{q}_0$ is the probability of choosing deterministically the most promising edge, q is a measure in range of [0, 1] and R is a randomly selected neighbor node.

Once a forward ant finds the required destination, a return ant is generated which retraces the path of the forward ant back to the source. The return ant then updates the value of pheromone at each intermediate node according to following reinforcement learning rule:

$$\tau\_ij = (1{-}e).\tau\_ij + \varDelta\tau\_ij \tag{4}$$

where the backward ant came from neighbor $j$ to node i, e is the rate of pheromone evaporation. Pheromone evaporation is a function of time and allows the system to forget the old information, search new paths and also avoid convergence to premature-optimal solutions by encouraging exploration of edges not yet visited. $\Delta\tau$ij is the amount of pheromone deposited with typically following form:

$$\varDelta\tau_{ij} \ \ = {}^{K}\!/_{f(c)} \tag{5}$$

where K > 0 is a constant. $f(c)$ is the cost function which can serve as a metric of hop counts from current node to destination, the delay of finding a destination, the available bandwidth of the link or the energy consumption of each node along the way. In general this amount of deposited pheromone depends on the quality of the solution which is found by a specific ant. In our model the security metrics, which affect the amount of deposited pheromone, are explained in the following system description part.

## 5 System descriptions

We consider a mobile ad hoc environment, in which all nodes perform four main processes to authenticate the destination node in order to have a secure communication. These processes are: public-private key generation and certificate issuing, certificate chains discovery, public key authentication by certificate verification, certificate chains trust updating. The following shows the details of each process.

### 5.1 Public-private key generation and certificate issuing

First each node creates its public key and corresponding private key locally. Then all neighboring nodes issue public key certificates for each other. If node A, based on its knowledge believes that a given public key $\mathrm{PK_B}$ belongs to a given node B, node A has to issue a certificate for node B and signs it with its private key $\mathrm{Pr_A}$, to show its assurance of the binding of identity B, $\mathrm{ID_B}$, to its related public key $\mathrm{PK_B}$. Each node saves the public key certificates it issues for others in its repository. For every node in the certificate repository there is a confidence level of trust that shows to which extent that node issues correct and not mismatched certificates. Figure 4 is an example of public key certificate generation for the nodes who are in each other's radio range.

Table 1 shows the certificate table (CT) in which every node stores the certificates issued by neighboring nodes. Each entry in CT corresponds to one public key and each

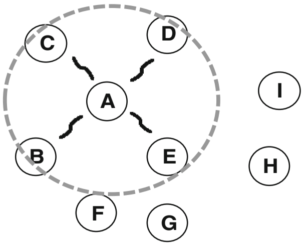

**Fig. 4** Certificate issuing for neighboring nodes located in the radio range

column shows the belief of each neighboring node to a certain public key.

Each node also has a table to store trust values of its neighboring nodes. Since this value presents the pheromone we name the table a trust-pheromone table (Table 2).

### 5.2 Certificate chain discovery

Our model is a reactive evidence distribution scheme. Ants are sent out only when a certain certificate is required. We assume that during the key generation and certificate issuing step, trust relationships have been established between nodes and their neighbors. In our proposed scheme the certificate chain discovery process contains of two phases:

- Forward phase or certificate request phase in which source node S sends out several ants to explore the path to obtain the public key of the destination node, D (Fig. 5). S sends forward ants in form of a certificate request packet to nodes that it directly trusts as a message like: $S \rightarrow B$: $\{cert\_req\}$.

An ant at each intermediate node, e.g. *B*, chooses, the next node e.g. *F*, according to formula (2), in which $\tau_{BF}$ is the pheromone which represents the referral trust that node *B* has on relaying node *F*. We assume that with the probability of $q_0$, neighbors with pheromone value more than the trust threshold should be chosen. $\eta_{ij}$ is the functional trust that node *i* has in general toward node *j* in providing the authentication service. In this work we assume that all nodes are trustable in making the authentication service available. The process is continued until the forward ant arrives at destination.

**Table 1** Certificate table of node A

| Certificates | Neighbors | | | |
|---|---|---|---|---|
| | B | C | … | E |
| $PK_i$ | $Cert_{B\rightarrow i}$ | $Cert_{C\rightarrow i}$ | | $Cer_{E\rightarrow i}$ |
| … | | | | |

**Table 2** Trust-pheromone table of node A

| Trust-pheromone | Neighbors | | | |
|---|---|---|---|---|
| | B | C | D | E |
| Pheromone | $t_{AB}$ | $t_{AC}$ | $t_{AD}$ | $t_{AE}$ |

- Backward phase or certificate reply phase in which the destination sends backward ants towards the source node. After receiving a forward ant at the destination, D in Fig. 5, it generates a backward ant and sends it to its previous node in the forward path, e.g. node F in Fig. 6. This backward ant is the certificate reply message: $D \rightarrow F$: $\{cert\_rep\}$.

A backward ant retraces exactly the path of the forward ant back to the source. Through returning of the backward ant from the destination to the source, each intermediate node adds certificates into reply packet and provides a chain of certificates (Fig. 6). For example, node F searches for the certificate of the destination, $Cert_{F\rightarrow D}$, in its certificate repository, adds it to the reply packet and sends it to B: $F \rightarrow B$: $\{cert\_rep, Cert_{F\rightarrow D}\}$

Node B receives this reply packet from F, searches for the certificate of F, adds it to backward ant and sends it to S as following message:

$$B \rightarrow S : \{cert\_rep, Cert_{F\rightarrow D}, Cert_{B\rightarrow F}\}$$

Finally the source node receives the backward ants from the destination, computes the trust value of the chains, using formula 1 and recovers the public key of the destination. The source node inserts the certificate chains and their corresponding trust values in its certificate chain table.

The source node will send out some limited number of ants to account for potentially lost backward ants, due to topology changes.

### 5.3 Public key authentication by certificate verification

After reception of several certificate chains the source node investigates the correctness of each certificate chain by verifying each certificate using the public key of its immediate next hop in the chain. For example we assume S receives SBFD as certificate chain (Table 3). Then it uses the public key of the

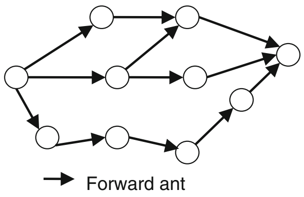

**Fig. 5** Forward ants carry certificate request

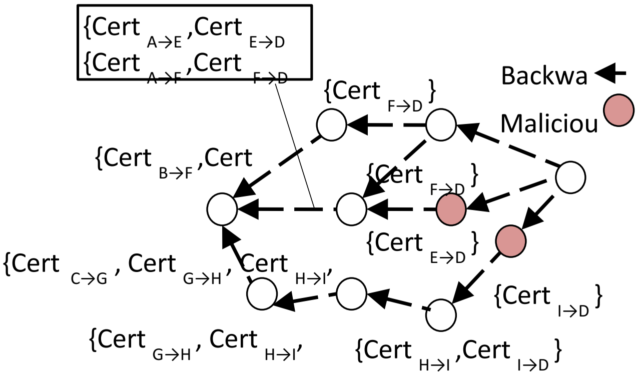

**Fig. 6** Backward ants carry certificate reply

node B, $PK_B$, to verify the first certificate in the certificate chain, $Cert_{B \to F}$. When verification of $Cert_{B \to F}$ is successful, the source node retrieves the public key of F, $PK_F$, through certificate $Cert_{B \to F}$ and then verifies the next certificate in the chain, $Cert_{F \to D}$. In case of having any discrepancy during verification phase, the certificate chain will be discarded and all nodes along the certificate chain will be punished.

If S detects no inconsistent certificates, it rewards all the nodes along the certificate chain and regards the maximum received trust value as trust value of the D's certificate:

$$t_{SD} = Max_{i \in CCh(S \to D)}\left(reliability_i\right) \tag{6}$$

where $CCh\ (S \to D)$ is the list of certificate chains which S receives after request for the public key of D. The reliability of the chain depends on the number of ants that traverse that path and the trust value of every intermediate node. It is calculated as follows:

$$reliability_i = ant_K\ \%.t_{SD} \tag{7}$$

Where $ant_K\ \%$ is the percentage of ants that traverse along this chain and $t_{SD}$ is the trust value of the chain calculated as formula 1.

### 5.4 Certificate chains trust update

Trust updates will occur in the three following situations:

**Table 3** Certificate chain table of a node

| Certificate chain | Destination | Trust value |
|---|---|---|
| SBFD | D | $t_1$ |
| SAED | D | $t_2$ |
| SAFD | D | $t_3$ |
| SCGHID | D | $t_4$ |

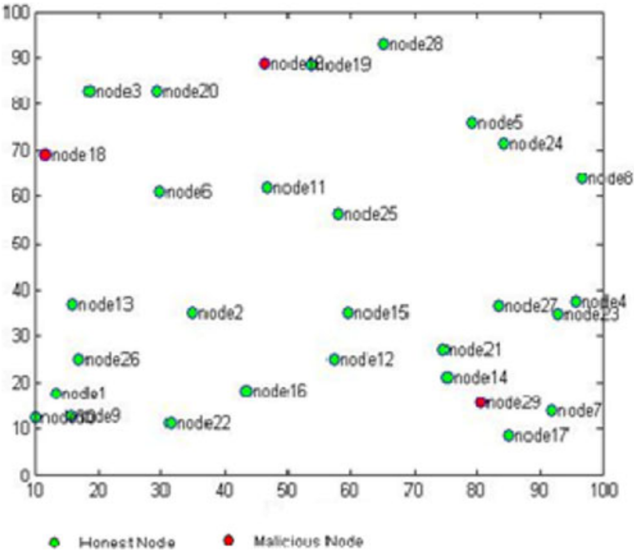

**Fig. 7** Example of a one tenth scaled scenario consisting of 10 % malicious node

**General local updating** In each intermediate node, if there is no mismatch information received by backward ants from its neighbors, the pheromone entry of the neighbor node, from where backward ant came from, will be updated as following:

$$t_{ij} = (1-e).t_{ij} + \ dt_{ij} \tag{8}$$

$$dt_{ij} = e.\left(1 + d_{jD}\ \ .(1.e).\left(1-t_{ij}.\eta_{ij}\right)\right).t_{ij} \tag{9}$$

where $e$ is the pheromone evaporation value. By $dt_{ij}$ in formula (8) we give the opportunity to edges with lower values of pheromone to recover faster. $d_{jD}$ is the distance, hop count, between j and the destination.

In our work we assume that every node only updates the trust value of its neighbor and the trust value of the new neighbors will be initialized with the trust threshold value.

**Table 4** Simulation parameters

| Parameter | Values |
|---|---|
| Number of nodes | 30 |
| Environment | 1000 × 1000 m$^2$ |
| Radio range | 250 m |
| Mobility model | Random waypoint |
| Speed | {0,2,10} m/s |
| MAC layer | 802.11 |
| Number of ants | {5, 10,15, 20, 25, 30, 35} |
| Trust threshold | 0.5 |
| e | 0.1 |
| $q_0$ | 0.98 |
| $\alpha = \beta$ | 1 |
| $\eta_{ij}$ *for* $\forall$ *i,j* | 1 |

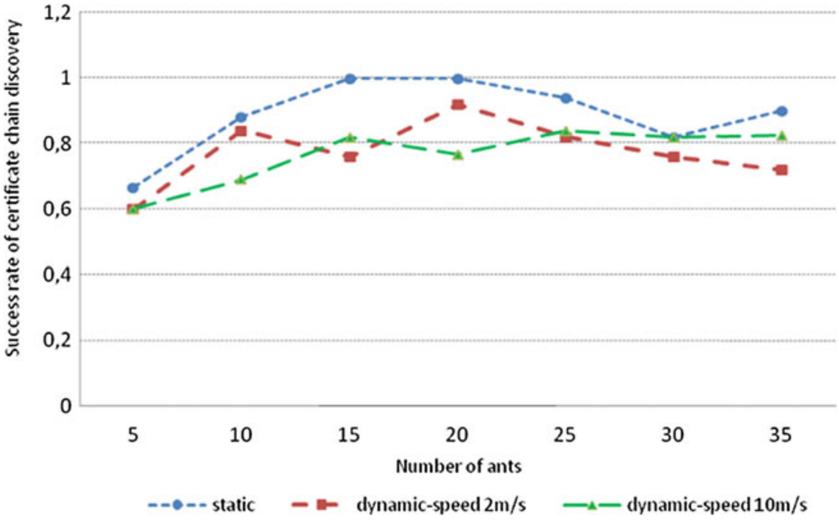

**Fig. 8** Success rate of certificate chain discovery versus varying number of ants

**Punishment** In case of observing any mismatch in certificate chains, node S analyzes the received certificate chain to identify Sybil nodes. S classifies the malicious nodes that offer confidentiality values for a certificate which is far from the opinion of the norm of the nodes in certificate chains. As this observation analysis is out of the scope of this paper, we suppose that the malicious nodes are already identified by the source node (e.g. node E and I in Fig. 6). In this case, as punishment, the source node reduces the trust level of its neighbors who led to the malicious node in the certificate chain by evaporating the pheromone of the edge between S and those neighbors (e.g. A and C in Fig. 6). Node S also has the responsibility of notifying A and C about the malicious nodes.

$$t_{ij} = t_{ij} - \omega.e.t_{ij}.dp, dp = \frac{1}{D_{PD}} \tag{10}$$

Weight $\omega$ is the trust value of a node toward its notifier neighboring node (e.g. $t_{AS}$ in Fig. 6). This weight is equal to 1 if the notifier itself is also the punisher. $D_{PD}$ is the distance factor (hop count) between punished (P) node and destination node (D), which can be obtained through the certificate chain. The longer this distance the less is the punishing amount.

All next nodes in the chain (e.g. A and C) will also punish their next immediate neighbor in the path towards the destination.

**Rewarding** Every node will reward its next hop along the reliable certificate chain and update its trust value asfollow:

$$t_{ij} = (1-e).t_{ij} + e.\left(1 + reliability_i.\eta_{ij}\right).t_{ij} \tag{11}$$

where $reliability_i$ is the reliability of the certificate chain. It shows the edges with higher pheromone value are more rewarded than those with lower value.

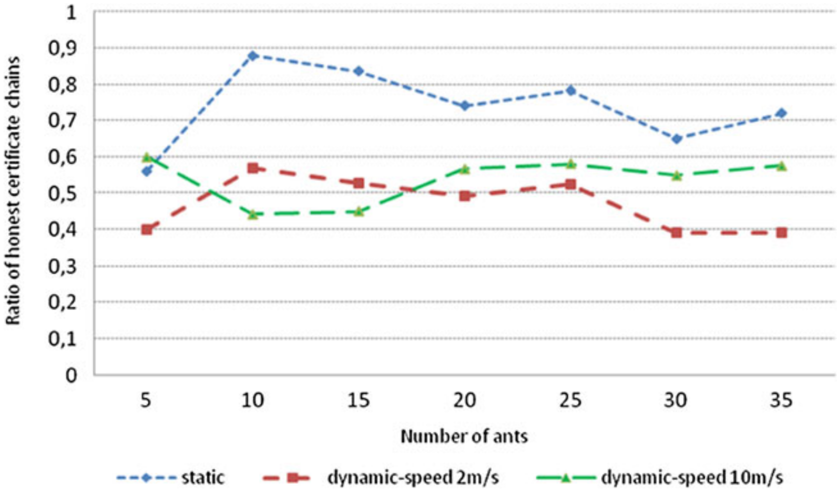

**Fig. 9** Ratio of reliable certificate chains versus different number of ants

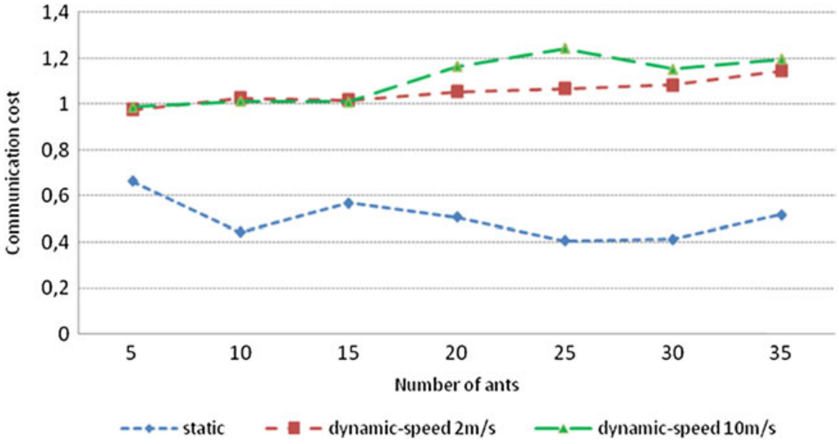

**Fig. 10** Communication cost of an authentication request against different number of ants

## 6 Simulation and results

### 6.1 Simulation environment

In order to investigate the efficiency and robustness of our self-organized authentication mechanism in an ad-hoc environment we used Matlab [21] for simulation. We use the MAC layer protocol 802.11 with data rate of 5 Mbps. 30 Nodes are randomly placed in 1000 × 1000 $m^2$, square area. Communication is in one hop distance with the radio range of 250 m. Random waypoint mobility is considered as mobility pattern with maximum speed of 0 m/s (static network), 2 and 10 m/s with a pause time of 3 s. At the beginning of the simulation, the nodes' uncertainty about their neighbors' trustworthiness is at the maximum level. Therefore in the initialization phase, the trust values between each node and its neighbors are considered equal to the trust threshold value. Here we consider this value equal to 0.5. Due to this initial value all neighbors of a node can be potentially chosen in initial requests, since the nodes environment is unknown.

During the simulation and application of pheromone updating, the trust values of malicious nodes converge to a value less than the threshold while the trust values of honest nodes approach a value higher than the threshold. Consequently, the pheromone updating phase reduces the uncertainty so that neighbors with higher trust values will be chosen during the certificate chain discovery phase.

Each run proceeds within a maximum of 600 s and in several iterations. We assume each iteration five requests for public keys of random honest destinations are generated. Requests are generated sequentially. After finishing each request and obtaining the public key of the destination, 5000 packets of data are transferred; and the pheromone values are updated. In our simulation, a public key certificate is considered to be 512 bytes in length, according the concept that encrypted data with a key size of 2048-bits should be secure. For the result in each iteration we consider the average of all five requests. After each round the updated pheromone values lunched to the network and new random requests are created. Figure 7 shows a scenario consisting of 10 % malicious nodes. In Table 4 the simulation parameters are summarized.

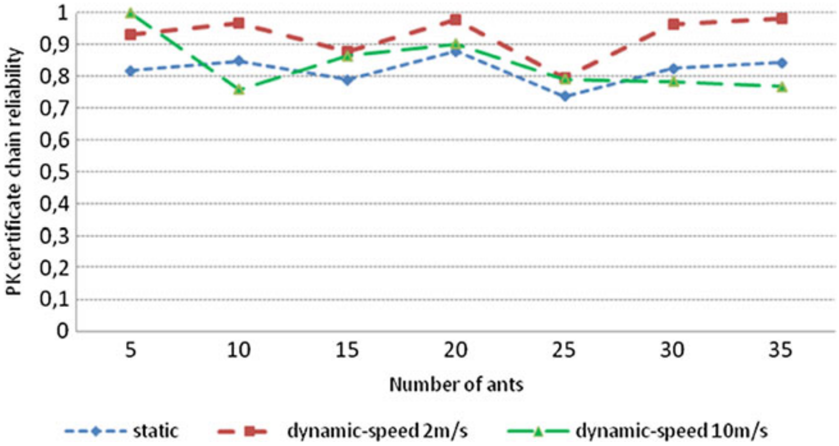

**Fig. 11** Certificate chain reliability versus different number of ants

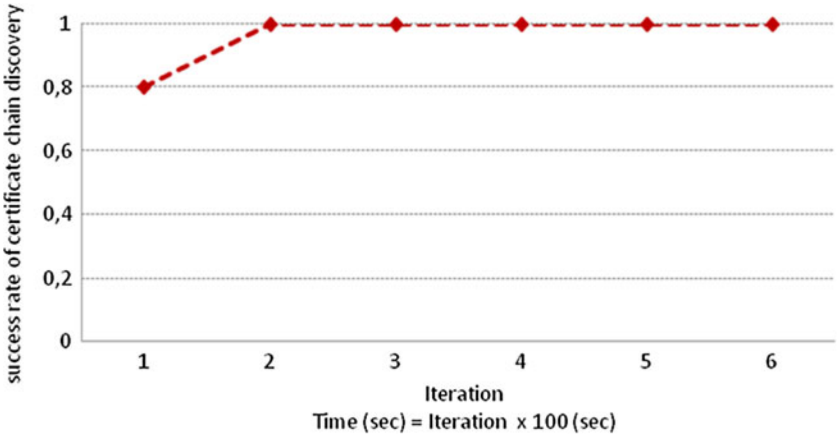

**Fig. 12** The success rate of certificate chain discovery through simulation time

### 6.2 Performance evaluation

In order to evaluate the performance of the proposed scheme we have selected the following three metrics:

- Success rate of certificate chains discovery: the percentage of requests for which the requester successfully obtains the public key certificate. This is the number of requests which obtained correct certificates over the total number of requests.
- Ratio of reliable certificate chains: this is the number of received certificate chains that succeed in the verification phase via the source node, to the number of transmitted ants that have found the destination.
- Reliability of the certificate chain: the average of the reliability of certificate chains for each request.
- Communication cost: we consider the average end-to-end delay during certificate chain discovery as a communication cost metric. It includes all delays at MAC and physical layer.

During the chain discovery phase, the backward ant waits for a duration of 1 s, to account for possible topology changes. When a forward ant arrives at the destination and completes its certificate chain, it will try to follow the same path backwards to the source node. After the waiting period the backward ant either takes the hop to the same node it came from, in case the node is still in communication distance or it is considered lost. For each forward ant, the source node waits for corresponding backward ants for a timeout period. We consider this time as a 0.2 s in our simulation. The source node has a three forward ants retransmission limit, when a timeout occurs. In case of a timeout, that may be caused by topology change, the destination node also sends at the maximum three backward ants.

In the first step of the performance evaluation we investigate the effect of number of ants sent out for each authentication request from source to destination, on the predefined metrics. Simulation runs have been performed for following different numbers of ants: 5, 10, 15, 20, 25, 30 and 35. In this step we assume that 10 % of the whole network is malicious nodes. The simulation results, of any change in number of ants, are averaged over all iterations in one run and also averaged over several runs.

Figure 8 shows that increasing the number of ants from 5 to 20, the success rate of certificate chains will increase. We consider the fluctuation in the diagram for mobile network with speed 2 m/s and 10 m/s, at the point with ant number 15 and 20 as because of the random nature of the scenario. However this rate will decrease after applying more than 20 ants in the static network and mobile

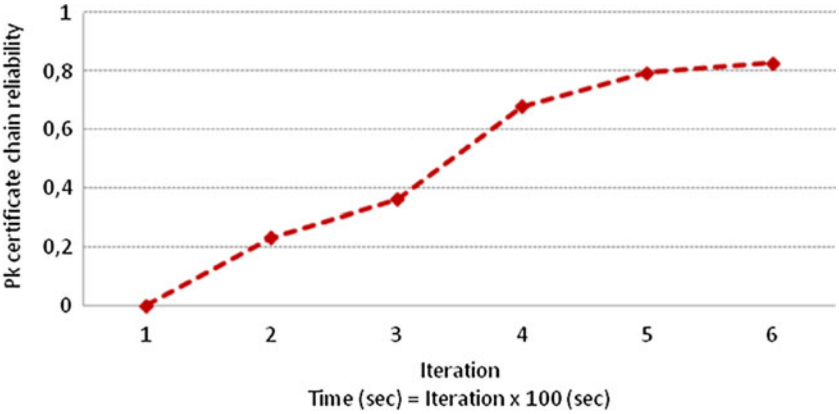

**Fig. 13** Evolution of public key certificate chain reliability through simulation time

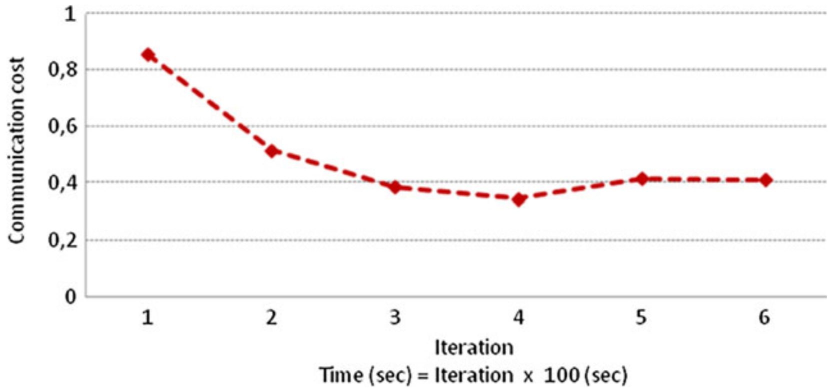

**Fig. 14** Communication cost through simulation time

network with 2 m/s; and it stays unchanged by dynamic network with speed of 10 m/s. Therefore the result shows that continuous increase of ants does not increase the success rate of certificate chains implicitly, in both static and dynamic networks.

Figure 8 also states that the success rate of certificate chain discovery is reduced for mobile networks compared to static networks. We reason that more backward ants get lost due to topology changes.

Figure 9 shows the ratio of the reliable certificate chain with varying numbers of ants. In both static and dynamic networks, increasing the numbers of ants for each authentication request does not increase the ratio.

The ratio of reliable certificate chains in static networks compared to mobile networks is higher. Again we reason the higher backward ant loss.

Further we investigate the effect of the number of ants on the average communication cost of authentication requests (Fig. 10). This value is considered as end-to-end delay of control and data packets necessary for an authentication request. It considers all dropped or successful packets reaching the destination. The results show that the average communication cost for each request, in a static network, declines with the number of ants. More ants can find the requested public key faster. However end-to-end delay in dynamic networks is not reduced because of the topology changes. Topology changes cause loss of more ants during the certificate chain discovery phase.

Figure 11 shows the maximum certificate chain reliability with varying numbers of ants. The maximum reliability of obtained public key certificate chains in networks with mobile nodes is higher than in static networks. Node mobility reduces uncertainty about the environment; because moving nodes have the opportunity to encounter new neighbors and consequently gather more knowledge about new nodes behavior.

The investigation of the effect of number of ants shows that increasing the number of ants is not a necessity for getting better results. For the remaining simulations we therefore use 20 ants for each request, since it reasonably fits the performance metrics.

Nodes in our self-organized authentication mechanism have no knowledge about their neighbors' behavior at start time. Therefore an evolutionary system is needed to gather information about the behavior of the other nodes, the public key signers, in the network. We present this process of self-learning along the period of simulation time, in our ant-based public key authentication model.

For this step we investigate the value of different metrics for each time interval. Like in the previous scenario we generate five random requests in each iteration (time intervals). The duration for each time interval is set to 100 s. We apply 20

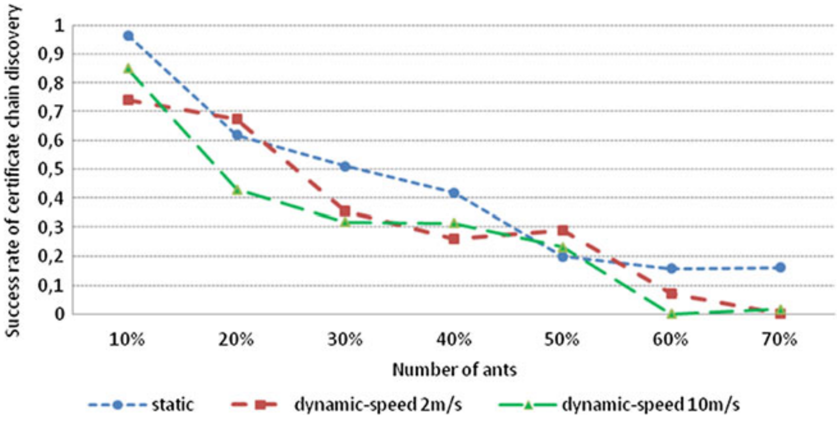

**Fig. 15** Success rate of certificate chain by increasing the percentage of malicious nodes in the network

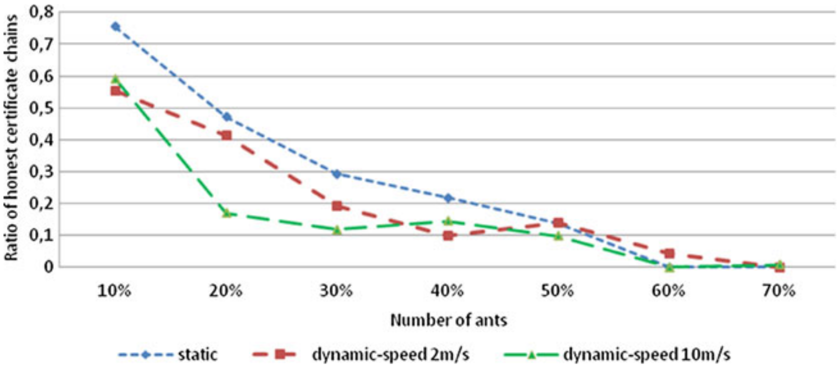

**Fig. 16** Ratio of reliable certificate chains in case of increasing percentage of malicious nodes in the network

ants for each request. The number of malicious nodes is 10 % of all nodes in the network, and nodes are static.

Clearly, in first iterations, because of having no knowledge about the trustworthiness and untrustworthiness of the neighboring nodes, the probability of choosing a malicious node along the certificate chain is higher. Therefore the success rate of correct certificate chains discovery might be lower than in the higher iterations; where trust values are updated. Figure 12 shows that in our scenario only in the first time interval, some public key certificate requests do not succeed.

Starting from the second iteration, the success rate of certificate chains discovery is always one; however Fig. 13 shows that the reliability of certificate chains is increasing through the time. It is stated that, when time passed, through updating pheromone values nodes with higher pheromone values are chosen as public key signer through a certificate chain.

By incentive mechanisms, the pheromone value of trustworthy nodes will be increased. Consequently in higher iterations most of the ants converged to the certificate chain with a higher trust value. Therefore the time of exploring honest certificate chains is reduced.

Figure 14 shows the reduction of the communication cost when simulation time is passing. In the third step we investigate the performance of the proposed scheme in case of increasing the number of malicious nodes for both static networks and networks with mobile nodes. To explore the effect of mobility on our proposed scheme we run simulations for mobility with a speed of 2 m/s and for 10 m/s. In all scenarios the source node sends out 20 ants for each public key request.

We increase the number of malicious nodes from 10 to 70 %. Figure 15 shows that the success rate of certificate chains in case of having near 20 % malicious nodes is above 0.5. The reduction of the success rate of certificate chains in networks with mobile nodes could result from the fact that more backward ants get.

Figures 15 and 16 also demonstrates the ratio of reliable certificate chains which is reduced by increasing the number of malicious node in the network. However it shows acceptable results comparing static network with a network of mobile nodes. Although mobility causes dropping backward ants by motion, however a node can encounter new neighbors and reduce its uncertainty about the environment by updating its pheromone table.

In the chain discovery phase nodes with trust value higher than threshold value are chosen, therefore when the number of malicious signer nodes is increased, the possibility of finding correct certificate chains is decreased and it leads to having less end-to-end delay.

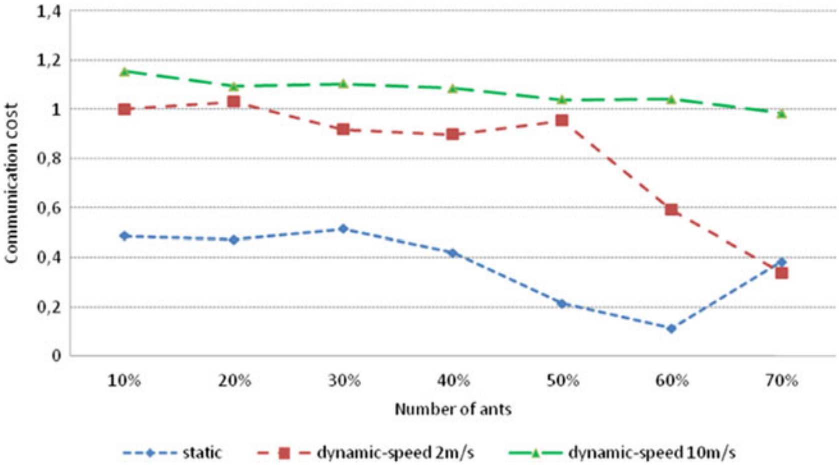

**Fig. 17** Communication cost for different percentage of malicious nodes

Figure 17 shows the decrease of communication cost when the number of malicious nodes is increasing. The cost of communication in networks with mobile nodes is higher than in the static network. The reason is the lost of ant by topology change or waiting time for finding a neighbor.

In Fig. 15 success rate of certificate chain in static network is about 0.2, despite of having 70 % malicious node in the network. This leads to increasing the communication cost around 70 % malicious nodes for static network in Fig. 17.

According to the obtained results, the proposed bio-inspired self-organized authentication mechanism is suitable for both static and mobile ad-hoc networks.

## 7 Conclusions

In this work, we proposed a robust self-organized public key management scheme for mobile ad-hoc environments based on an ant colony system. Due to the lack of central authority in these environments achieving an acceptable level of security in the autonomous authentication process is a difficult problem. Our scheme is able to authenticate and obtain the public key of a target node successfully despite of malicious relay nodes. Traces of pheromones represent the trust level of nodes throughout the certificate chain.

We simulated our proposed scheme and explored its performance by varying the number of ants. The results show that a high number of ants is not absolutely necessary for achieving better performance. Different metrics in the certificate chain discovery phase have been measured. Our model shows considerable robust behavior against malicious nodes. We also conclude that our proposed model is suitable for static networks and networks with low to high mobility.

As future work we plan to investigate the scalability of the network by increasing the number of nodes. We also plan to make our proposed self-organized mechanism be able to cope with colluding malicious nodes and detect the certificate chains contain of Sybil nodes with multiple identity.

## References

1. Hashmi S, Brooke J (2008) Authentication mechanisms for mobile ad-hoc networks and resistance to sybil attack. Proc Second Int Conf Emerg Secur Inf Syst Technol IEEE Comput Soc
2. Mármol FG, Pérez GM (2009) Security threats scenarios in trust and reputation models for distributed systems. Comput Secur 28(7): 545–556
3. Cordon O, Herrera F, Stützle T (2002) A review on the ant colony optimization metaheuristic: basis, models and new trends. Math Soft Comput
4. Zimmermann P (1995) The official pgp user's guide. MIT Press
5. Capkun S, Buttyan L, Hubaux J-P (2003) Self-organized public-key management for mobile ad hoc networks. IEEE Trans Mob Comput 52–64
6. Li R, Li J, Liu P, Chen H (2006) On-demand public-key management for mobile ad hoc networks: research articles. Wirel Commun Mob Comput 6:295–306
7. Dahshan H, Irvine J (2010) A robust self-organized public key management for mobile ad hoc networks. Secur Commun Netw 16–30
8. Dahshan H, Irvine J (2009) On demand self-organized public key management for mobile ad hoc networks. IEEE 69th Vehicular Technology Conferece
9. Jøsang A, Ismail R, Boyd C (2007) A survey of trust and reputation systems for online service provision. Decis Support Syst 618–644
10. Marti S, Garciamolina H (2006) Taxonomy of trust: categorizing P2P reputation systems. Comput Netw 472–484
11. Kamvar S, Schlosser M, Garcia-Molina H (2003) The Eigentrust Algorithm for Reputation Management in P2P networks. In: Proceedings of the 12th international conference on World Wide Web (WWW '03). Budapest, Hungary: ACM
12. Mármol FG, Pérez GM, Skarmeta AFG (2009) TACS, a trust model for P2P networks. Wirel Pers Commun 1:153–164
13. Buchegger S, Le Boudec JY (2004) A robust reputation system for P2P and mobile ad-hoc networks. In: Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems
14. Michiardi P, Molva R (2002) Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In: Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security 107–121
15. Ganeriwal S, Srivastava MB (2004) Reputation-based framework for high integrity sensor networks. In: Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, ACM: Washington DC, USA 66–77
16. Boukerch A, Xu L, Khatib E (2007) Trust-based security for wireless ad hoc and sensor networks. Comput Commun 30(p):2413–2427
17. Wang W, Zeng G, Yuan L (2006) Ant-based reputation evidence distribution in P2P networks. In: Proceedings of the Fifth International Conference on Grid and Cooperative Computing. IEEE Comp Soc 129–132
18. Jiang T, Baras JS (2004) Ant-based adaptive trust evidence distribution in MANET. In: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops. IEEE Comput Soc 7:588–593
19. Jøsang A, Golbeck J (2009) Challenges for robust of trust and reputation systems. In: Proceedings of the 5th International Workshop on Security and Trust Management, Saint Malo, France
20. Douceur JR (2002) The sybil attack. In: Revised Papers from the First International Workshop on Peer-to-Peer Systems. Springer-Verlag 251–260
21. MathWorks: http://www.mathworks.com/products/matlab/