Bio-inspired Self-organized Public Key Authentication Mechanism for Mobile Ad-hoc Networks

Parisa Memarmoshrefi, Roman Seibel, and Dieter Hogrefe

Institute for Computer Science, Telematics Group, University of Göttingen
Goldschmidtstrasse 7, 37077 Göttingen, Germany
{memarmoshrefi,seibel,hogrefe}@cs.uni-goettingen.de

Abstract. In mobile ad-hoc networks (MANETs), where there is no centralized authority to provide security, trust and reputation mechanisms are applied to maintain security by identifying trustworthy and untrustworthy nodes. However, traditional authentication mechanisms are infeasible 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 path in certificate chains gathered by ants and can identify and prevent certificate chains with individual or colluding malicious nodes.

Keywords: public key authentication, security threat in trust and reputation systems, ant colony optimization, MANETs.

1 Introduction

Mobile ad-hoc networks are multi-hop wireless networks without any infrastructure which are used in different applications, such as civilian or military applications and emergency rescues. In environments where cooperation is unavoidable providing security services for communication is an essential issue. Since authentication is the most important and the 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 between nodes is necessary, in which the keys are transmitted in a secure way over basically insecure channels. A framework of trust relationships is required to be built for authentication purposes in the 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 MANETs collectively perform the task of a CA; 3- self

J. Suzuki and T. Nakano (Eds.): BIONETICS 2010, LNICST 87, pp. 375–386, 2012.

CA systems which are based on web of trust. This model allows nodes to become an individual CA, generate their own keying material and issue public key certificates for their own and for others base on their knowledge. Through a certificate, the binding of a node's identity to its corresponding public key is proven by a digital signature of the issuer. Each node maintains a local certificate repository, and performs the public key authentication via chain of certificates.

However there still exist security threats in this trust model. A number of security threats are presented in [3] which in general could be applied in trust and reputation systems. Therefore one of the most important subjects is identifying and coping with dishonest misbehaving nodes along the certificate chains who try to cheat other nodes into believing in false node-public key bindings.

To mitigate the problem, we propose on-demand, trust-based public key management based on ant colony systems [4]. The dynamic nature of ad hoc networks, caused by the mobility of nodes and the changing behavior of nodes, makes ant colony optimization an appropriate choice for a trust model. 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 toward the target node. The responsibility of the ants is to find the most trustworthy certificate chain. At the same time, through building a certificate chain, the ants leave traces of pheromone on the path representing a trust level of the path. Despite of misbehaving nodes each node can make a suitable decision about obtaining the public key of a target node.

The rest of the paper is organized as follows: Section 2 represents some related works. Section 3 includes trust model and security threats of 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 conclusion and future works.

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 [5] certificates are mainly stored in a centralized certificate repositories. [6] proposes a self-organized public key management where certificates are stored and distributed by the nodes. The main problem of this scheme is large overhead for storing the approximate global certificate graph. To solve this problem, authors in [7] proposed a solution. They designed an on-demand public key management. 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 discovered between source node and destination node. Recently another solution is proposed [2] which is based on the existence of a web of trust.

On the other hand, many trust and reputation systems have been proposed, for dealing with malicious behavior, in many different domains such as human social networks, e-commerce [8], peer-to-peer networks [9][10][11], mobile ad-hoc networks CONFIDANT [12] and CORE [13] and sensor networks [14][15].

TACS [11], which was helpful for our model, AntRep [16] and [17] are trust models using a bio-inspired algorithm of the ant colony systems in order to provide guarantee of network resources availability and trustworthiness. In these 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).

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 $Cert_{i\rightarrow j}$ denotes the certificate that i signed with its private key, Sig_i, to show the binding of node j's identity, ID_j, and its corresponding public key PK_j. In addition to ID_j and PK_j , the data structure $Cert_{i\rightarrow j}$ contains trust value or confidentiality, C, and validity time, T. We consider this web of trust a certificate graph G(V, E), whose set of vertices, V, represents public keys and the set of edges, E, represents 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.

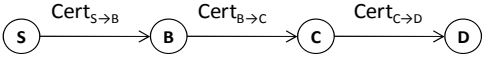

Fig. 1. Certificate Chain

In our example the certificate chain from S to D is $\{Cert_{S\rightarrow B}, Cert_{B\rightarrow C}, Cert_{C\rightarrow D}\}$. 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 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 recommendation 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 [18]. 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.

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.

$$t_{SD} = \prod_{k=1}^{k=n} t_k \quad (1)$$

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 source and 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 raises by not participating in the authentication process or issuing a false certificates with an incorrect binding of a key to an identity. Impersonating another node is an example of functional misbehavior. A malicious node i may issue a certificate that binds the identity of another node, ID_j, to its public key, PK_i, and signs it with its private key Pr_i. The aim of the malicious node is eavesdropping a messages sent to j. Another example is binding the public key of node k, PK_k , to ID_j; although it should be bound to 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. One of the important threats of this kind is the Sybil attack [19], where a malicious node generates several keys and identities, binds the IDs to corresponding public keys and issues certificates for them. In this case the malicious node can use these nodes to issue false certificates. As the false certificate is signed by many Sybil nodes, it could be considered as a correct certificate to non Sybil nodes.

The aim of our proposed model is a self-organized authentication mechanism which enables defense against these two types of misbehavior. In this paper we concentrate on misbehaving nodes who try to defect the authentication service by disseminating false information.

4 Ant Colony Optimization

In a system 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. They remember the visited nodes they pass, and deposit 'pheromone' on them. Ants are attracted to paths with higher pheromone concentration. When an ant wants to move from starting node S toward a destination it chooses one of the neighboring nodes of S, i, with the probability defined by following transition rule:

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

where τ_{Si} is the pheromone deposit on the edge between S and i, η_{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 α and β are the weights for balancing between deposited pheromone and goodness value of the edge respectively.

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

$$r = \begin{cases} \underset{j\epsilon N(S)}{\operatorname{argmax}}[\tau_{Sj}]^{\alpha}.[\eta_{Si}]^{\beta} & \text{if} \quad q \le q_0 \\ R & \text{otherwise} \end{cases} \tag{3}$$

where r is the next chosen node by an ant in its next movement, 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:

$$\tau_{ij} = (1-e).\tau_{ij} + \Delta\tau_{ij} \tag{4}$$

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 $\Delta\tau_{ij} = {}^{K}/_{f(c)}$. $K > 0$ is a constant. $f(c)$ is the cost function which serves 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. The security metrics are explained in the system description part.

5 System Descriptions

We consider an ad hoc environment, in which all nodes perform five main processes: public-private key generation and certificate issuing, certificate chains discovery, public key authentication by certificate verification, certificate chains trust updating and certificate revocation. 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 PK_B belongs to a given node B, node A has to issue a certificate for node B and sign it with its private key Pr_A , to show its assurance of the binding of identity B, ID_B, to its related public key PK_B. Each node saves the public key certificates it issues for others and certificates issued to it 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 2 is an example of public key certificate generation for the nodes who are in radio range of each other.

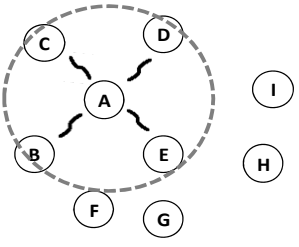

Fig. 2. Certificate issuing for neighboring nodes located in the radio range

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 certificate and each column shows the belief of each neighboring node to a certain certificate.

Table 1. Certificate Table of Node A

Certificates	Neighbors			
	B	C	D	E
$Cert_i$	$Cert_{B\rightarrow i}$	$Cert_{C\rightarrow i}$	$Cert_{D\rightarrow i}$	$Cer_{E\rightarrow i}$
…				

Each node also has a table to store trust value of its neighboring nodes. Since this value presents the pheromone we name the table a trust-pheromone table (table2).

Table 2. Trust-Pheromone Table of Node A

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

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. We also assume that source node S wants to obtain the public key of destination node, D (Figure 3). S sends out several ants to explore the path to the required certificate. Ants in each node, e.g. *i*, choose their next state, e.g. *j*, according to formula (2), in which τ_{ij} is the pheromone and represents the referral trust that node *i* has on relaying node *j* on obtaining the public key of destination node, D. η_{ij} is the functional trust that node i has in general toward node *j* in providing authentication service. In this work we assume that all nodes are trustable in making the authentication service available.

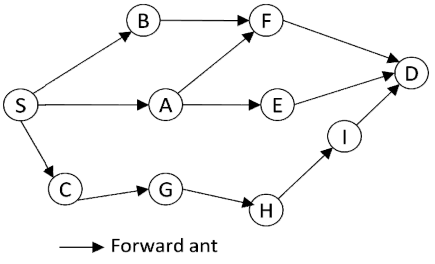

Fig. 3. Forward ants carry certificate request

Forward ant stores the certificate path. Once a forward ant finds the required certificate, a backward ant is generated who retraces exactly the path of the forward ant back to the source. Through returning from destination to source, the backward ant adds the certificates of the intermediate nodes in its packet and provides a chain of certificates (figure 4).

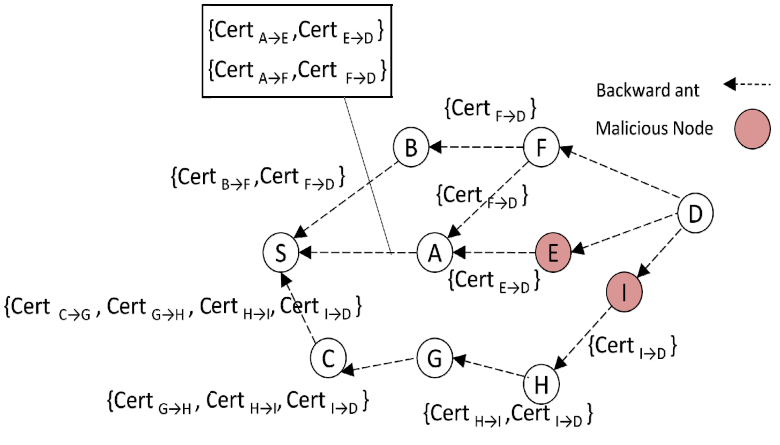

Fig. 4. Backward ants carry certificate reply

Finally when the source node receives the backward ants from destination, it computes the trust value of the chain, using formula 1, and inserts the certificate chains and their corresponding trust values into the certificate chain table.

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

In case of any topology changes during a backward ant tries to find the way back to the source, source node should send out some other ants toward target node.

5.3 Public Key Authentication by Certificate Verification

When the source node receives different chains of certificates, it verifies the ID-key binding of the destination which is contained in certificate chains. If S detects no conflicting certificates (e.g. certificates contain the same identity of D but with different public keys), it regards the maximum received trust value as trust value of the D's certificate:

$$t_{SD} = Max_{\,i \in CCh(S \to D)}(t_i) \tag{5}$$

where $CCh(S \to D)$ is the list of certificate chains which S receives by requesting for certificate of D. However, considering the existence of misbehaving nodes, S may receive mismatched destination certificates through different certificate chains.

5.4 Certificate Chains Trust Update

Trust updates will occur in three following situations:

5.4.1 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:

$$t_{ij} = (1 - e).t_{ij} + dt_{ij} \tag{6}$$

$$dt_{ij} = e.\left(1 - t_{ij}.\eta_{ij}\right) \tag{7}$$

where e is the pheromone evaporation value. By dt_{ij} in formula (7) we give the opportunity to edges with lower values of pheromone to recover faster.

5.4.2 Punishing: 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 figure 4). 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 figure 4). Node S also has the responsibility of notifying A and C about the malicious nodes.

$$t_{ij} = t_{ij} - \omega . e . t_{ij} . dp \quad , \quad dp = \frac{1}{D_{pm}} \tag{8}$$

Weight ω is the trust value of a node toward its notifier neighboring node (e.g. t_{AS} in figure 4). This weight is equal to 1 if the notifier itself is also the punisher. D_{pm} is the distance factor (hop count) between punished (p) node and malicious node (m), which can be obtained through the certificate chain. The longer this distance the less is the punishing amount. The punished node who continues to act maliciously as a consequence will be isolated and not further used in the authentication process.

Nodes A and C also have to punish their next neighbors in the path leading to the malicious nodes; Otherwise they will be classified as potentially malicious nodes.

5.4.3 Rewarding: Source node S updates the trust value of every node along the most reliable certificate chain as follows:

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

where t_{Max} is the highest trust value corresponding to the most reliable certificate chain. It shows the edges with higher pheromone value are more rewarded than those with lower value.

5.5 Certificate Revocation

If a node believes that in a certificate it issued, the binding of ID to public key of the target node is no longer valid or the trust value is less than the trust threshold, t_{th} , it can revoke that certificate. When a node receives a certificate revocation, it compares the trust value of sender of the message, t_{is}, and the trust value of the nodes its certificate is claimed to be revoked, t_{it}:

$$t_{it} = t_{it} - t_{is} \tag{10}$$

If t_{it} is lower than the trust threshold, it deletes the revoked certificate, otherwise the certificate will still be used.

6 Simulation and Results

We assume that trust relationships have been established between each node and its neighbors. Matlab is used for simulation. The simulation parameters are as follows. 30 nodes are randomly placed in the 100 x 100 meter square form area. Nodes are variables and the radio range of each node is 30 meter. Also we consider the following values for parameters: $\alpha = \beta = 1,\ e = 0.1, q_0 = 0.9, \eta_{ij} = 1\ for\ \forall\ i,j.$

The simulation proceeds in rounds. Each round the updated pheromone values lunched into the network. In each round five requests are made. In each request one of the four randomly chosen nodes requests the public key certificate of one of two random destination nodes. The numbers presented in each iteration are averaged over all five requests.

First we consider all requests are to be the same and between a fixed pair of source and destination nodes. Figure 5 shows the trust value of received certificate chains in case of increasing the percentage of malicious nodes in the network. It is shown that after some iteration the network learns the chains of trustable nodes that lead to the certificate of the destination node. The result shows that the model is capable of choosing the trustworthy nodes to get the public key certificate of the destination node despite of up to 60% malicious nodes. The reduction of the reliability of the certificate chain in case of having 30% malicious nodes is because of the location of some interconnected nodes. If these nodes, which connect one part of the network to another part, are malicious no reliable path could be found between these two parts of the network.

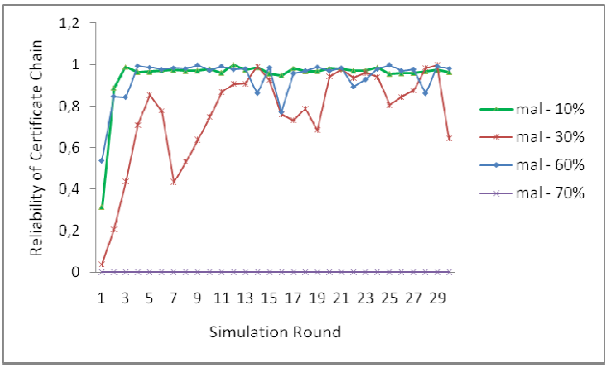

Fig. 5. Reliability of certificate chain with different percentage of malicious nodes

In the second experiment we made five different requests and compare the success rate with different amounts of malicious nodes. The success rate shows the percentage of requests for which the requester successfully obtains the public key certificate. It is the number of correct certificates obtained over the total number of requests each round.

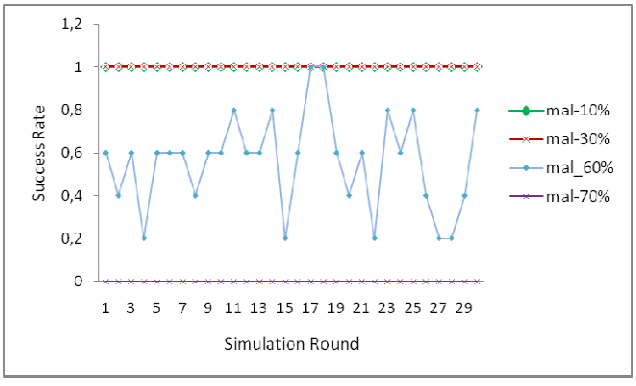

Fig. 6. Comparison of success rate with different percentage of malicious nodes

Figure 6 shows success rate in case of up to 30% malicious nodes in network, is constantly 1. By increasing the number of malicious nodes to 60% the average success rate is nevertheless more than fifty percent. All requests are completely failing if the amount of misbehaving nodes reaches 70%, which means 21 nodes out of 30 nodes.

7 Conclusions and Future Works

In this work, we proposed a robust self-organized public key management scheme for ad-hoc environments based on ant colony systems. This scheme is able to authenticate and obtain the public key of a target node successfully in spite of malicious relay nodes. Traces of pheromones represent the trust level of nodes throughout the certificate chain.

As our future work we aim at clustering data gathered by ants in order to identify Sybil nodes. Using ants, the source node has the opportunity to gather information about the public key of the destination through different chains containing different recommending nodes. Therefore by analyzing the recommendations, the source node can identify Sybil nodes who try to pretend to be different nodes and disseminate false information. We intend to implement our proposed scheme by a network simulator and explore its performance and consider further parameters such as node mobility and its effects.

References

1. Hashmi, S., Brooke, J.: Authentication Mechanisms for Mobile Ad-Hoc Networks and Resistance to Sybil Attack. In: Proceedings of the Second International Conference on Emerging Security Information, Systems and Technologies. IEEE Computer Society (2008)
2. Dahshan, H., Irvine, J.: A Robust Self-Organized Public Key Management for Mobile Ad hoc Networks. Security and Communication Networks, 16–30 (2010)

3. Mármol, F.G., Pérez, G.M.: Security Threats Scenarios in Trust and Reputation Models for Distributed Systems. Computers & Security 28(7), 545–556 (2009)
4. Cordon, O., Herrera, F., Stützle, T.: A Review on the Ant Colony Optimization Metaheuristic: Basis. Models and New Trends. Mathware & Soft Computing (2002)
5. Zimmermann, P.: The Official PGP User's Guide. MIT Press (1995)
6. Capkun, S., Buttyan, L., Hubaux, J.-P.: Self-Organized Public-Key Management for Mobile Ad Hoc Networks. IEEE Transactions on Mobile Computing, 52–64 (2003)
7. Li, R., Li, J., Liu, P., Chen, H.: On-Demand Public-Key Management for Mobile Ad hoc Networks: Research Articles. Wireless Communications & Mobile Computing 6, 295–306 (2006)
8. Jøsang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems, 618–644 (2007)
9. Marti, S., Garciamolina, H.: Taxonomy of Trust: Categorizing P2P Reputation Systems. Computer Networks, 472–484 (2006)
10. Kamvar, S., Schlosser, M., Garcia-Molina, H.: The Eigentrust Algorithm for Reputation Management in P2P networks. In: Proceedings of the 12th International Conference on World Wide Web (WWW 2003). ACM, Budapest (2003)
11. Mármol, F.G., Pérez, G.M., Skarmeta, A.F.G.: TACS, a Trust Model for P2P Networks. Wireless Personal Communications 1, 153–164 (2009)
12. Buchegger, S.: Le Boudec. J.Y.: 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 (2004)
13. Michiardi, P., Molva, R.: 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, pp. 107–121 (2002)
14. Ganeriwal, S., Srivastava, M.B.: Reputation-based Framework for High Integrity Sensor Networks. In: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, pp. 66–77. ACM, Washington DC (2004)
15. Boukerch, A., Xu, L., Khatib, E.: Trust-based Security for Wireless Ad hoc and Sensor Networks. Computer Communications 30, 2413–2427 (2007)
16. Wang, W., Zeng, G., Yuan, L.: Ant-based Reputation Evidence Distribution in P2P Networks. In: Proceedings of the Fifth International Conference on Grid and Cooperative Computing, pp. 129–132. IEEE Computer Society (2006)
17. Jiang, T., Baras, J.S.: Ant-Based Adaptive Trust Evidence Distribution in MANET. In: Proceedings of the 24th International Conference on Distributed Computing Systems Workshops, vol. 7, pp. 588–593. IEEE Computer Society (2004)
18. Jøsang, A., Golbeck, J.: Challenges for Robust of Trust and Reputation Systems. In: Proceedings of the 5th International Workshop on Security and Trust Management, Saint Malo, France (2009)
19. Douceur, J.R.: The Sybil Attack. In: Druschel, P., Kaashoek, M.F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002)