A Self-Organized Mechanism for Thwarting Malicious Access in Ad Hoc Networks

Natalia Castro Fernandes, Marcelo Duffles Donato Moreira, and Otto Carlos Muniz Bandeira Duarte
GTA/COPPE - Universidade Federal do Rio de Janeiro - Rio de Janeiro, Brasil

***Abstract*—This paper introduces a self-organized mechanism to control user access in ad hoc networks without requiring any infrastructure or a central administration entity. The proposed mechanism authenticates and monitors nodes with the so-called controller sets, which are resistant to the dynamic network membership. The analysis shows that the proposed scheme is robust even to collusion attacks and provides availability up to 90% better than proposals based on threshold cryptography. The performance improvement arises mostly from the controller sets autonomy to recover after network partitions.**

I. Introduction

Ad hoc networks do not rely on a physical infrastructure or a central administration entity. Indeed, a different user controls each node and therefore security becomes a major issue to keep collaborative message forwarding working. To restrict undistinguished node access in regular networks, two complementary approaches are used: access control and authentication. When we can authenticate nodes and identify malicious actions, the network is able to punish malicious nodes and reward the cooperative ones. In ad hoc networks, however, both access control and authentication are challenging, because they are usually based on centralized mechanisms. Accordingly, ad hoc networks demand self-organized mechanisms based on distributed administration and nodes with equivalent functions, providing high availability even on network partitions.

In this paper, we propose ‘A Controller-node-based Access-Control mechanIsm for Ad hoc networks’ (ACACIA). ACACIA is a self-organized public-key management and monitoring system that dismisses any trusted central authority or fixed server. In ACACIA, all nodes play an equal role and the proposed mechanisms guarantee high availability even if network membership and topology are highly dynamic. Our mechanism provides both authentication and access control that are suitable for ad hoc networks. ACACIA is based on two features: delegation chains and controller nodes. The delegation chain is used to control network access without a centralized administration. As a result, all users are responsible for allowing new users to join the network. Users that allow a malicious user to join are punished in order to keep the network secure. The controller nodes, introduced in this paper, manage certificates and also monitor and punish nodes. Each node in the network is controlled by a dynamic controller set composed of randomly chosen nodes. The random selection provides a distributed access control with a different controller set for each node. Indeed, every node belongs to controller sets of other nodes. Therefore, our scheme avoids nodes with special functions because all nodes have the same duties of certificating and monitoring nodes. The controller set actions are threshold ruled to prevent colluding attacks. An action is taken only if the majority of the set agrees on this action.

Once ad hoc network membership is highly dynamic, our controller nodes adapt to network context by changing ACACIA parameters according to the network size. Besides, a controller set is regenerated whenever a membership change affects this controller set. Since the controllers are dynamically chosen and based on voting, they provide autonomy and availability to ACACIA. As the controller sets monitor and punish nodes, ACACIA is safe against attacks and controls node access. The performed evaluations show that ACACIA availability is up to 90.7% greater than the threshold cryptography-based proposals on network partitions.

This paper is structured as follows. Related research concerning authentication are reviewed in Section II, while system model assumptions are presented in Section III. The proposed scheme is detailed in Section IV and the proposal evaluation is in Section V. Finally, we draw conclusions in Section VI.

II. Related Work

The typical certificate authority (CA) model is inadequate for ad hoc networks because it requires infrastructure and a central administration. Hence, the challenge for these distributed networks is to manage keys and control network membership. Zhou and Haas proposed the distribution of the certification authority through the network in a threshold fashion [1]. The idea is to distribute the CA responsibilities among a specialized group of nodes with a (k, m) threshold scheme. Each of these special nodes receives a share of the master private key of the CA. Therefore, a certificate is only issued if at least k out of the m specialized nodes agree about this task. This approach, however, can degrade the CA availability, because k out of m nodes may not be accessible to nodes all the time, especially during network initialization and partitions. Other approaches using threshold cryptography have been proposed [2], [3], but the need for an administrator to manage membership or select and configure a special group of nodes persists in all of these proposals.

In the web-of-trust authentication model, if user A trusts user C, which signed user B public key, then A also trusts B. As this process develops, a web of trust is built through certificate chains. Based on this idea, some proposals for authentication in ad hoc networks were done [4], [5]. In these proposals, nodes maintain repositories of certificates issued by themselves or received from other trusted nodes. When a node A needs to authenticate node B, it looks for a certificate chain from node A to node B in its repository. If this chain is found, node A signs B's certificate and exchanges repositories with B. One of the main problems of these proposals is the network initialization, when most nodes have a limited certificate repository and are not able to find certificate chains. Another issue is that nodes are free to generate fake identities due to the absence of access control and then subvert the authentication scheme [6].

Our proposal better fulfills the ad hoc network requirements than the previous schemes because it avoids a central administrator for controlling network access or configuring special purpose nodes, like in threshold cryptography proposals. Both threshold cryptography and web-of-trust proposals have problems in the initialization due to a low number of nodes in the network. In ACACIA, the self-adaptation of the parameters to network conditions and the self-management of the controller nodes favor both initialization and network partitions and still guarantee security. Moreover, the randomized selection of the controllers reduces the probability of collusion attacks.

III. System Model

A. Network and Adversary Models

We assume that a group of people with a common interest or relationship is motivated to create a private social network. Any user that belongs to the delegation chain is called an authorized user and has free access to the private network. We further assume an ad hoc network in which node mobility and link outages cause frequent network partitions. For sending a message, we consider four types of communication: broadcast, unicast, multicast, and flood. Broadcast represents a one-hop transmission, unicast is a transmission from one node to other, multicast is a transmission from one to n nodes, and the flood is a transmission to the whole network.

We consider as adversaries users who try to damage the network. Non-authorized users can only act as passive attackers, unless they subvert the proposed scheme and obtain certificates as authorized users. We also consider as adversary authorized users that create or discard messages to hazard the network or to save energy power. Both non-authorized and malicious authorized users are on the focus of the proposed scheme.

IV. The proposed Scheme

Our access control scheme authorizes, authenticates and monitors users. To accomplish these functions, the proposed scheme uses a delegation chain and controller nodes. Delegation chains are hierarchical structures used to delegate a resource among a group of entities authorized by the resource owner, though sometimes not all the group of entities are know by the resource owner [7]. We use delegation chains based on the users' relationship to distribute the access control. If the social network is hierarchical, this relationship hierarchy is trivially used to define the root of the delegation chain. In our scheme, only users that belong to the delegation chain can authorize other users to join the private network. We introduce the parameter *Max. Descendents* in the authorization, which is described in Fig. 1(a), to restrict malicious access by specifying the maximum number of descendents that each user can have. This parameter indicates how much the father trusts in the child he accepted to join the delegation chain. If the father has a strong relationship with the child, he specifies a high maximum number of descendents.

A. The Key Idea: The controller sets

The main feature of ACACIA is the dynamic and self-managed controller set. ACACIA associates to each node in the network a controller set, which issues certificates, monitors nodes, and excludes malicious nodes from the network. The controller nodes that belong to the controller set of a specific node control the network access of this specific node. Hence, the controller set is randomly selected to avoid that the controlled node manipulates the controller set membership.

Since ad hoc network membership is highly dynamic due to joining/leaving nodes and network partitions, a static controller set does not fit well. As a consequence, we propose a self-management mechanism, described in Sections IV-B3 and IV-B4, which redefines the controller sets whenever there is a change in their elements.

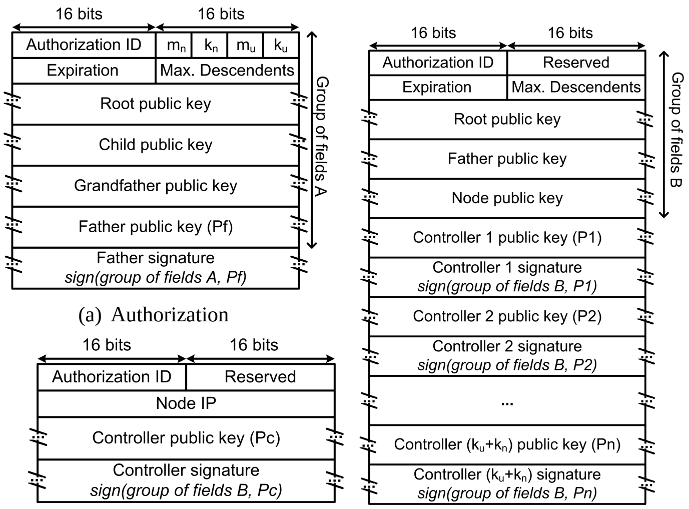
In ACACIA, the controller set of each node is divided into two subsets, called node-controller set (N_c), which monitors node behavior, and user-controller set (U_c), which controls the delegation chain. These set sizes, given by $|N_c| = m_n$ and $|U_c| = m_u$, are specified by the delegation chain root in the authorization. All the controller nodes issue certificates. As any node can belong to a controller set, the decisions are always taken based on voting. A certificate is issued only if at least k_n node controllers and k_u user controllers agree that the controlled node is authorized to access the network. Hereafter, we assume $k_n \geq \lfloor m_n/2 \rfloor + 1$ and $k_u \geq \lfloor m_u/2 \rfloor + 1$, which means that both thresholds are set majorities to guarantee the system reliability against malicious nodes.

The node-controller set specific functions are node monitoring and malicious node exclusion. If a node in the network detects a malicious action, it sends a notification to the node controllers of the malicious node. Each node controller runs a trust system and votes to exclude or not the controlled node from the network according to the received reports.

The user-controller set guarantees the delegation chain validity. Hence, they verify and store authorizations of the monitored nodes. In addition, they control the number of children of the monitored nodes. After receiving an authorization, a user can join the network only if his father has already participated in the network. If this condition holds, the new user's controller sets contact the father's user-controller set to verify the validity of the authorization. After a user joins the network, his user-controller set is formed and is automatically maintained even if the user leaves the network. Whenever the user-controller set changes, the user controllers exchange information so that the new user controller knows all data about the monitored node. As a consequence, the presence of the father is not required to a child join the network.

B. ACACIA Detailed Description

1) New Node Access: This mechanism, described in Fig. 2, allows users to obtain an IP address, the node public/private keys and the corresponding certificate. First, the joining user obtains offline an authorization, depicted in Fig. 1(a), with a previously authorized user based on his social relationship and on the delegation chain. After that, the joining user needs the allocated IP list to choose an address and know his controller sets. Hence, after overhearing a Hello-message from a node, the joining user asks for the allocated IP list with a New-message. The Hello source node replies this message with the broadcast of a List-message, which contains the allocated address list in a compact form. The broadcast is needed because the joining user does not have an IP yet. After obtaining the allocated IP list, the user chooses a pair of public/private keys and selects the first p bits of the public key as his IP address suffix. This process is repeated until the user finds a pair of keys associated to an available address, which we call node public/private key.



(a) Authorization

(b) Partial certificate.

(c) Complete certificate.

Fig. 1. Certificates in ACACIA.

After choosing an available IP address, the joining user calculates his controller sets based on the received allocated IP list, as we explain in Section IV-B4. Next, he multicasts a message with his authorization and his node public key to his controllers and his father's user controllers. He signs this message with the child public key specified in the authorization to prove he owns the authorization and to associate the authorization with the chosen IP. When the joining user's controller nodes receive this message, they store the received data and waits for a period T_V for the answer of the father's user controllers. This is needed to validate the delegation chain. If the father's user controllers guarantee the father belongs to the delegation chain, then the child also belongs to the delegation chain. Hence, the father's user controllers verify the authorization signature and check if the father can have one more child. If these conditions hold, the father's user controllers multicast the Validation-message to the joining user's controllers. After receiving at least k_u out of the m_u Validations from father's user controllers, the joining user's controllers flood the network with a Partial Certificate, depicted in Fig. 1(b), and also with their list of allocated IPs. Provided that the joining user receives k_u out of the m_u Partial Certificates from his user controllers and k_n out of the m_n Partial Certificates from his node controllers, his admission in the network has been authorized and he generates a Complete Certificate, described in Fig. 1(c). Finally, the joining user floods an Allocation-message to notify the network that he has a certificate and the IP address is allocated. Every node that receives the Allocation-message and at least k_n out of the m_n Partial Certificates from the joining user's node controllers allocates the new address.

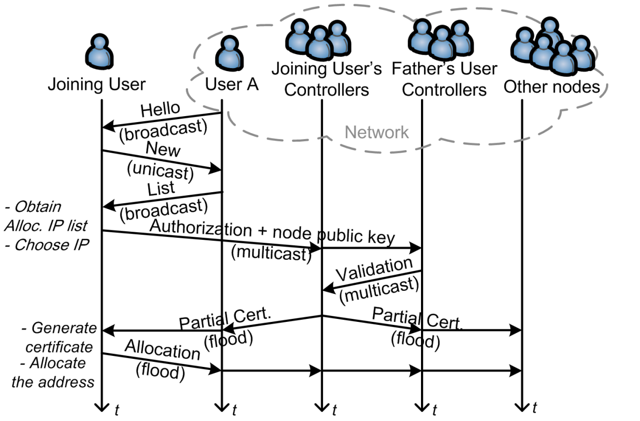


Fig. 2. Message exchanges over time in the new node access mechanism.

After obtaining the complete certificate, the joining user can take part with the network and becomes member of other nodes controller sets. Furthermore, this user will start to send periodical Hello-messages informing the hash of the root public key, which identifies the delegation chain the user belongs to, and also the hash of the allocated IP list, which is used in the identification of partition merging events.

2) Node Monitoring: In ACACIA, all nodes run a bad behavior detection system. Each node monitors its neighbors and notifies the neighbor's node controllers whenever a bad behavior is detected. Based on this notification and on a trust system, the controller nodes exclude monitored nodes. Therefore, every node excludes a malicious node after receiving at least k_n out of m_n votes from the node controllers of the malicious node. Also, all the descendent nodes of the malicious node must be excluded by their controller nodes, which is achieved hop-by-hop in the delegation chain with the votes of the controllers. Any bad behavior detection system (BBDS) and any trust model [8] can be used with ACACIA.

3) Network Partitions and Membership Management: The controller sets detect the departure of the nodes based on a frequent contact with the controlled node. Hence, each node sends a probe packet with period T_T to its controllers. If a controller does not receive any packet in a period $p_t{\cdot}T_T$, where p_t is the lost packet threshold, then this controller announces that the monitored node has left the network. When at least k_u user controllers and k_n node controllers agree that the node has left, all nodes exclude that node from the allocated IP list. As a consequence, this mechanism guarantees that the allocated IP list is always updated when a partition is formed or after a node leaves the network.

The partition merging causes a different challenge, in which there are a lot of new nodes that will not execute the new node access mechanism. To solve this, ACACIA identifies partition merging events using the hash of the allocated IP list in the Hello-messages. Indeed, partitions have different allocated IP lists, and, consequently, different hashes of the list. If two neighbors have different hashes of the list, they are probably in different partitions and must merge their lists. Hence, an ACACIA node floods its allocated IP list whenever there are more than T_U seconds since the last list update and the hash of the list received in its neighbor's Hello is different from its own hash of the list. When a node receives a message with a list, it checks if there is more than T_U seconds since the last list update. If this condition holds, the node merges its list with the received one, accepting as present all the absent nodes which are considered as present in the received list.

After a partition merging, the user controllers of a node which were in different partitions exchange information to assure that the number of issued authorizations is correct. Besides, the controllers verify if the node was excluded in the other partition due to bad behavior or to the exclusion of an ascendant in the delegation chain.

The partition merging mechanism also solves message losses. If Partial Certificates or an Allocation-message of a specific node is lost, some nodes allocate the node IP while other nodes do not. Hence, there will be differences on the hash of the allocated IP list, which will start this mechanism until all the nodes have the same allocated IP list.

4) Selecting Controller Sets: The selection of the nodes in each controller set is made via hash functions to guarantee that the controllers are uniformly distributed among the nodes in the network. The set selection must be executed every time a node joins or leaves the network to maintain the controller sets updated. Algorithm 1 shows how to select the controller sets. In this algorithm, $x \in \{user, node\}$, L_n is the ordered list of the active nodes, $node$ is the IP of the controlled node, and C_x is the controller list. Also, the function ϕ is a simple hash function applied i times over key. In this algorithm, we consider that $L_n = \{l_1, \cdots, l_n\}$ is circular. That is, after $l_n \in L_n$, we find $l_1 \in L_n$. The user-controller set has size m_u, while the node-controller set has size m_n.

During network initialization, m_u and m_n are higher than the number of nodes, N. In this case, we choose $m_n = m_u = N - 1$. Therefore, to increase system availability, we reduce the needs for security in network initialization and we reestablish security thresholds as the network membership increases. When $max(m_n, m_u) < N - 1$, the controller sets are formed by m_u and m_n nodes as specified in the authorization. These values will not change unless the network membership is reduced so that the number of nodes is not enough to create the controller sets of each node.

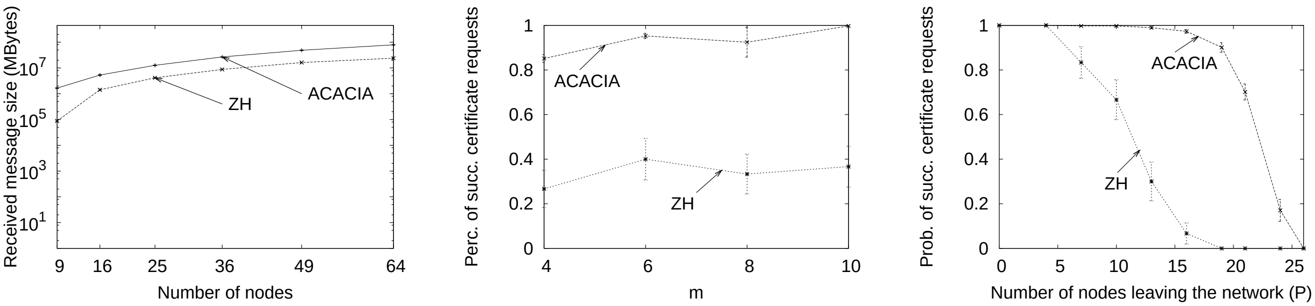
The selection of the controller sets is seeded by the input key. The algorithm selects a controller candidate by applying the key in function ϕ. The result of ϕ selects a position in the list of allocated addresses, L_n. Then, we check if this candidate has already been selected as a controller. If the node is not a controller, it is selected as part of the controller set C_x. Instead, we select the next node in the circular list L_n that is not a controller to be the new controller. Once the node controllers, responsible for the monitoring system, must be known by all nodes, the key for the node-controller set is the node IP. Hence, every node in the network can calculate the node-controller set of any node. For the user controllers, the key is the father public key, which is available in the certificate. The use of the father public key is important to avoid nodes from using the same authorization to obtain many IPs simultaneously and execute a Sybil attack. The choice of different IPs results in different node-controller sets. The use of the father public key, however, guarantees the choice of same user-controller set for the same authorization independent of the chosen IP.

Algorithm 1: Controller set selection.

```
Input: $node$, $m_x$, $k_x$, $L_n = \{l_1, ..., l_n\}$, $key$
Output: $C_x = \{c_1, ..., c_{m_x}\}$, $m_x$, $k_x$
$N = size(L_n)$
if $(N - 1 \leq m_x)$ then
    $m_x = k_x = N - 1$
    $C_x = L_n - \{node\}$
end
if $(N - 1 > m_x)$ then
    $C_x = \{c_1 = -1, ..., c_{m_x} = -1\}$
    foreach $i = 1 : m_x$ do
        $T_x = C_x$
        $c_i = (\phi(key, i)\ mod\ N) + 1$
        while $c_i \in T_x$ do
            $c_i = min\{l \in L_n\ |l > c_i\}$
        end
    end
end
```

V. Simulations

We implemented ACACIA in the Network Simulator-2. The simulations model radio propagation using the Shadowing model and the Medium Access Control using the IEEE 802.11 model. As routing protocol, we used the Ad hoc On-demand Distance Vector routing protocol (AODV). These choices account for creating a model that closely matches a community network, using indoor parameters of commercial equipments. We evaluated the control traffic and the certificate distribution of ACACIA. We compare ACACIA with the protocol proposed by Zhou and Haas, which we call ZH. ZH is considered the main proposal for authentication in ad hoc networks, because it distributes the certification authority among a group of nodes. Both ACACIA and ZH are based on threshold values to reach to a decision. The certification authority (CA) is composed of m_t nodes in ZH and the threshold is given by $k_t = \lfloor m_t/2 \rfloor + 1$. For the purpose of a fair comparison, we choose the parameters of ACACIA as $m = 2 \cdot m_n = 2 \cdot m_u = m_t$. Consequently, both CAs have size m. We suppose the threshold for user-controller set, whose size is m_u, is given by $k_u = \lfloor m_u/2 \rfloor + 1$ and the threshold for node-controller set, whose size is m_n, is given by $k_n = \lfloor m_n/2 \rfloor + 1$. Instead we state differently, the other



(a) Overhead of control messages received by all the nodes.

(b) Simulation results for the obtained certificates after a partition of size $N_P = 13$.

(c) Percentage of obtained certificates after a partition of size N_P.

Fig. 3. Simulation results for ACACIA and ZH.

simulation parameters are in Table I. The field configuration is chosen as a square area in which nodes are randomly distributed and we consider a confidence level of 95% in the results.

TABLE I
SIMULATION PARAMETERS.

Description	**Value**
Number of nodes (N), density	52, 0.03 nodes/m^2
$m_t = 2 \cdot m_u = 2 \cdot m_n$	8
k_t, k_u, k_n	5, 3, 3
Probe packets interval (T_T), lost packet thres. (p_t)	2 s, 3
Validation time (T_V), list update interval (T_U)	1 s

It is expected that ACACIA control load is greater than ZH, because our protocol updates the certificate authority according to the network parameters and monitors nodes. Hence, Fig. 3(a) shows the control overhead after 50 s of simulation. The number of nodes impacts in both results due to the floods in the routing protocol and in ACACIA.

The network partition effect was also analyzed assuming a network with N_F nodes, in which N_P randomly chosen nodes leave the network simultaneously. After that, a group of N_L nodes join the network. In this configuration, all the ZH certificate authority nodes are among the first N_F nodes. Besides, the first N_F nodes of the ACACIA delegation chain correspond to the first N_F nodes joining the network. The probability of the last N_L nodes obtain the certificate shows the system availability after a partition of size N_P, which is depicted in Fig. 3(b) assuming $N_F = N_L = 26$ and $N_P = N_F/2 = 13$ nodes. The number of obtained certificates in ZH is almost the same for all $m = m_t$ values, because while m_t increases, $k_t = \lfloor m_t/2 \rfloor + 1$ also increases. For ACACIA, if $m = 4$, then $m_u = 2$, and $k_u = 2$, which means that the protocol has no redundancies on the controller sets. Therefore, the ability to recover user controllers after a partition is degraded, depending on the presence of the node to reestablish its controllers. For $m = 6$, 8, and 10, ACACIA availability increases because $m_u > k_u$ and the user-controller set can be automatically recovered after a partition. Fig. 3(b) shows that ACACIA outperforms ZH availability by up to 65%. Using the same configuration, we evaluated the partition size effect, varying the N_P, as shown in Fig. 3(c). In all the cases, ACACIA has a greater availability than ZH. ZH availability quickly decreases as more nodes leave the network, while ACACIA is more robust to the size of the partition. If 16 out of the 26 first nodes leaves the network in ZH, there is a probability of 6.6% that the system works for the last N_L nodes. In the same situation, the success probability in ACACIA is 97.3%.

VI. CONCLUSION

We proposed an access control mechanism, called ACACIA, which introduces a new strategy to distribute administrator duties among all the authorized users. By virtue of controller sets, ACACIA monitors node behavior and purges from the network non-cooperative nodes, which safeguards the ad hoc network performance. ACACIA is robust and self-adaptive to network conditions, even during network initialization and network partitions. The cost for flexibility and for the absence of infrastructure or a central administration is a higher message overhead when compared to systems based on threshold cryptography, due to the dynamic update of the controller sets. The analysis results bring out characteristics of ACACIA, such as a high availability even in the presence of partitions when compared to other mechanisms. Therefore, ACACIA is well-suited to ad hoc characteristics and is a feasible alternative for ad hoc networks applications that require security.

REFERENCES

[1] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, no. 6, pp. 24–30, 1999.

[2] J. Luo, J.-P. Hubaux, and P. T. Eugster, "DICTATE: Distributed certification authority with probabilistic freshness for ad hoc networks," *Trans. Dependable Secure Comput.*, vol. 2, no. 4, pp. 311–323, 2005.

[3] H. Luo, J. Kong, P. Zerfos, S. Lu, , and L. Zhang, "URSA: Ubiquitous and robust access control for mobile ad hoc networks," *IEEE/ACM Transactions on Networking*, vol. 12, no. 6, pp. 1049–1063, Dec. 2004.

[4] S. Capkun, L. Buttyán, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 25–64, Mar. 2003.

[5] J.-P. Hubaux, L. Buttyán, and S. Capkun, "The quest for security in mobile ad hoc networks," in *2nd ACM Intl. Symp. on Mobile Ad Hoc Networking & Computing (MobiHoc '01)*. ACM, 2001, pp. 146–155.

[6] S. Hashmi and J. Brooke, "Authentication mechanisms for mobile ad-hoc networks and resistance to sybil attack," in *Second International Conference on Emerging Security Information, Systems and Technologies (SECURWARE '08)*, Aug. 2008, pp. 120–126.

[7] D. Yao and R. Tamassia, "Compact and anonymous role-based authorization chain," *ACM Transactions on Information and System Security (TISSEC)*, vol. 12, no. 3, pp. 1–27, 2009.

[8] P. B. Velloso, R. P. Laufer, O. C. M. B. Duarte, and G. Pujolle, "Analyzing a human-based trust model for mobile ad hoc networks," *IEEE Symposium on Computers and Communications (ISCC'2008)*, pp. 1–6, Jul. 2008.