Self-Organized Authentication in Mobile Ad-Hoc Networks

Pino Caballero-Gil and Candelaria Hernández-Goya

Abstract: **This work proposes a new distributed and self-organized authentication scheme for mobile ad-hoc networks (MANETs). Apart from describing all its components, special emphasis is placed on proving that the proposal fulfils most requirements derived from the special characteristics of MANETs, including limited physical protection of broadcast medium, frequent route changes caused by mobility, and lack of structured hierarchy. Interesting conclusions are obtained from an analysis of simulation experiments in different scenarios.**

Index Terms: **Access control, authentication, cryptography, mobile ad-hoc networks (MANETs).**

I. INTRODUCTION

Services such as authentication, confidentiality, integrity, non-repudiation, availability, and access control are the main base for network security. Among all these facilities, authentication, which ensures the true identities of nodes, is the most fundamental one because other services depend fully on the correct authentication of communication entities.

Mobile ad-hoc networks (MANETs) may be described as autonomous networks formed by mobile nodes that are free to move at will. These networks have received increasing interest in the last years, partly owing to their potential applicability to many different situations, ranging from small, static networks that are constrained by power sources, to large-scale, mobile, and highly dynamic networks. Whilst conventional wired networks normally use a globally trusted certificate authority (CA) for solving the authentication problem, such a solution is not the best for MANETs. In fact, the authentication problem in MANETs is much more difficult to solve due to their characteristics such as the absence of a fixed infrastructure and centralized management, the dynamic nature and limited wireless range of nodes, the dynamic topology, frequent link failures and possible transmission errors [1], [2]. Also, since all nodes must collaborate to forward data, the wireless channel is prone to active and passive attacks by malicious nodes, such as denial of service (DoS), eavesdropping, spoofing, etc.

This work proposes a new distributed and self-organized authentication scheme for MANETs, which fulfils most requirements of this type of networks, including limited physical security, high node mobility and lack of infrastructure.

Manuscript received March 13, 2008; approved for publication by Heejo Lee, Division III Editor, June 22, 2009.

This work was supported by the Spanish Ministry of Education and Science and the European FEDER Fund under TIN2008-02236/TSI Project, and by the Agencia Canaria de Investigación, Innovación y Sociedad de la Información under PI2007/005 Project.

The authors are with the Departamento de Estadística, I.O. y Computación, Universidad de La Laguna, Spain, email: {pcaballe, mchgoya}@ull.es.

This paper is organized as follows. In Section II some existing solutions are briefly described. Section III provides an overview of the proposed scheme, including general aspects and notation. Specific details about the five principal elements of the architecture, i.e., network initialization, node insertion, access control, proofs of life, and node deletion are gathered in Section IV. The assumptions required by the proposed scheme and an analysis of its security are commented in Section V. Section VI provides a performance analysis developed under NS-2. Finally, some conclusions and open questions complete the paper.

II. RELATED WORK

In 1999 Zhou and Haas [3] suggested using threshold cryptography to secure MANETs. They proposed a distributed CA to issue certificates, but this idea is not applicable to ad-hoc groups since only selected nodes can serve as part of the certification authority, and contacting the distributed CA nodes in a MANET may be difficult. Luo *et al.* considered the same problem in [4] and Kong *et al.* in [5]. They proposed a set of protocols for ubiquitous and robust access control in MANETs, which allow every member to participate in access control decisions. Unfortunately, this scheme has been shown to be insecure in [6].

Another interesting identification paradigm that has been used in wireless ad-hoc networks is the notion of chain of trust [7], but it fails if malicious nodes are within the network. Another typical solution is location-limited authentication, which is based on the fact that most ad-hoc networks exist in small areas and physical authentication may be carried out between nodes that are close to each other. However, the location-limited authentication is not feasible for large, group-based settings.

Later, Kim *et al.* [8] developed a group access control framework based on a menu of cryptographic techniques, which included simple access control policies, such as static access control lists (ACLs), as well as admission based on the decision of a fixed entity: External (e.g., a CA or a trusted third party) or internal (e.g., a group founder). The main drawback of such a proposal is that those policies are inflexible and unsuitable for dynamic ad-hoc networks. For instance, static ACLs enumerate all possible members and hence cannot support truly dynamic membership, and admission decisions made by a trusted third party (TTP) or a group founder violate the peer nature of the underlying ad-hoc group.

Other authentication protocols that have been recently proposed for ad-hoc networks are the following. The work [9] based on the RSA signature conducts to the problem of public key certification. Another recent paper [10] provides a solution that works well, but just for short-lived MANETs.

In conclusion, we may say that the design of new schemes

that fulfil most requirements for this type of networks continues being considered an open question, and indeed is the main objective of this paper.

Here we propose a new architecture for authentication in ad-hoc networks called global authentication scheme for mobile ad-hoc networks (GASMAN), which is based on the established cryptographic paradigm of zero-knowledge proofs (ZKPs) [11]. Since the information sent while executing does not convey any secret related to the authentication process, ZKPs provide an elegant and fault-tolerant solution to node authentication in MANETs. As we will see in this paper, when comparing the GASMAN with existing proposals, several improvements are remarkable:

1. In the proposed scheme all nodes play exactly the same role. In particular, there are no selected nodes serving as CA and admission decisions are not made by a TTP or a group founder but by the nodes themselves.
2. The GASMAN has scalability and flexibility and is suitable for dynamic ad-hoc networks, thanks in part to that it is not based on any static structure such as ACLs.
3. The proposal is feasible for group-based and long-lived MANETs. A key factor to achieve it is the fact that it is not based on location-limited authentication.
4. Availability is guaranteed through insertion, deletion, and access control procedures.
5. Our architecture assures strong authentication to any legitimate node willing to join the network by using the ZKP implemented in the access control.
6. The GASMAN algorithms jointly with mobility help to reduce the time necessary for nodes to join and access the network in a timely manner.

Summing up, the main features of the proposal are the adaptation to the varying topology of the network, the open availability of broadcast transmissions, and the strong access control.

Up to now, very few publications have mentioned the proposal of authentication systems for ad-hoc networks using ZKPs. Two of them are [12] and [13], but none dealt with the related problem of topology changes in the network. Another recent ZKP-based proposal for MANETs related with the one proposed here was the hierarchical scheme described in [14], where two different security levels were defined through the use of a hard-on-average graph problem, but again no topology changes were considered. On the other hand, two works that may be considered the seed of this work are [15] and [16]. The main differences between the proposal of this paper and both references are the following: Definition of node life-cycle, analysis of possible attacks, description of necessary assumptions, provision of a larger example, more data about performance analysis, and a comparison with existent solutions.

III. BASICS AND NOTATION

With the term authentication, here we refer to verification of users' identities. Another important concept in this paper is availability, which involves making network services or resources available to legitimate users in such a way that the survivability of the network is ensured despite malicious incidences. The architecture proposed in this paper is intended both for authentication and for availability.

In particular, the protocol was designed as a strong authentication scheme for group membership since when a node wants to be part of the network, it has to be previously authorized by a legitimate node through a validation process of its identity against previously stored information by using cryptographic credentials. According to [17], in any group member authentication protocol it is necessary to provide robust methods to insert and to delete nodes, as well as to allow the access only for legitimate members of the group. For that reason, not only the ZKP used for access control is described, but also the update procedures associated to insertions and deletions are carefully defined. For instance, the procedure to delete nodes is only initiated once a node has been disconnected of the network for too long. The period of time after which the node is deleted is an important parameter (T) of the system here presented.

Note that in this paper strong authentication does not refer to multi-factor authentication [18] since we consider just one factor for the authentication process. Consequently, the proposal could be improved by adding more factors to the authentication process, but even in such a case the strength of the scheme would be always bound to the secrecy under which the factors are kept.

The access control described below is based on the general scheme of ZKP introduced in [19], when using the Hamiltonian cycle problem (HCP). A Hamiltonian cycle in a graph is a cycle that visits each vertex exactly once and returns to the starting vertex. Determining whether such cycles exist in a graph defines the HCP, which is NP-complete. Such a problem was chosen for our design mainly due to the low cost of the operations associated to the update of a solution. This is an important characteristic since in a highly dynamic setting such as MANETs these operations will be developed frequently. Anyway, there should be pointed out that similar schemes based on different NP-complete graph problems might be described. The only feature demanded to the chosen problems is that the solutions may be easily updated when small changes occur in the network. This is just the case of the vertex cover, independent set or clique problems, for instance.

One of the key points to assure the correct operation of GASMAN is the use of a chat application through broadcast that makes it possible for legitimate on-line nodes to send a message to all on-line users. Such an application allows publishing all the information associated to the update of the network. In order to provide integrity of chat information, the sender could sign a hash of the chat message, and even such a hash could be encrypted using a symmetric cipher with the shared secret key. On the other hand, although secrecy is not necessary for chat messages because they are useless for illegitimate nodes, it is required that only the on-line legitimate nodes can execute the chat application. Consequently, prior authentication of the users of the chat application is required. To solve this matter, the access control based on ZKP described in Section IV could be used.

The information received through the chat application during an interval of time must be stored by each on-line node in a FIFO queue. Such data should be stored by each on-line node, allowing in this way the updating of the authentication information not only to it but also to all the off-line legitimate nodes whose

access will be granted. Such a period is an essential parameter in the system because it states both the maximum off-line time allowed for any legitimate node, and the frequency of broadcasting the proofs of life. Consequently, such a parameter should be previously agreed among all the legitimate nodes of the network.

A generic life-cycle of a MANET has three major phases that are described below (see Fig. 1):

Initialization:

Each initial member of the original network will be securely provided, either off-line or on-line, with a secret piece of information. The knowledge of the secret network key will be used during access control in order to prove the node's eligibility for accessing to the protected resources or to offer service to the network. After completing this stage, the legitimate nodes are ready to actively participate in the network.

Access control:

The access control process allows a legitimate node to prove its network membership to an on-line node. These legitimate nodes must demonstrate knowledge of the secret network key by using a challenge-response scheme.

On-line session:

Once the legitimate node reaches an on-line state in the network, it gets full access to protected resources such as the chat application. At the same time, it may offer services such as the insertion of new nodes. There should be taken into account that the secret network key will be updated according to the network evolution. Hence, if a node is off-line for too long, its secret key will expire. In such a case, the legitimate node would have to be re-inserted by an on-line legitimate node.

Since in our proposal the secrecy of the network key is provided by the difficulty of the HCP, the number of on-line legitimate nodes is a crucial parameter. In consequence, as soon as the number of on-line legitimate nodes becomes too small (when comparing it with certain threshold parameter), the network termination is carried out and therefore, the life-cycle of the network ends.

A remarkable aspect of our proposal, which is shared with other previous proposals, is that no meaningful information may be stolen even if an adversary is able to read the whole information published through the chat application, or even if it eavesdrops the information exchanged between a legitimate prover and a legitimate verifier at the time of executing the access control protocol.

In the following, the basic notation used throughout the proposal is explained.

- $G_t = (V_t, E_t)$ denotes the undirected graph used at stage t of the network life-cycle.
- $v_i \in V_t$ represents both a vertex of the graph and a legitimate node.
- $n = |V_t|$ is the order of G_t, which coincides with the number of legitimate nodes.
- $N_{G_t}(v_i)$ denotes the neighbors of node v_i in the graph G_t.
- $\Pi(V_t)$ represents a random permutation over the vertex set V_t
- $\Pi(G_t)$ denotes the graph isomorphic to G_t built after applying permutation $\Pi(V_t)$.
- $c \in_r C$ indicates that an element c is chosen at random with uniform distribution from a set C.
- HC_t designates the Hamiltonian cycle used at stage t.
- $\Pi(HC_t)$ represents the Hamiltonian cycle HC_t in the graph $\Pi(G_t)$.
- $N_{HC_t}(v_i)$ denotes the neighbors of node v_i in the Hamiltonian cycle HC_t.
- S and A stand for the supplicant and the authenticator, respectively. This notation is used both while an insertion phase and while the execution of a ZKP-based access control are carried out.
- $S \rightleftharpoons A$ symbolizes when node S contacts A.
- $A \leftrightarrow S$: *Information* means that A and S agree on *information*.
- $A \overset{s}{\rightarrow} S$: *Information* means that A sends $information$ to S through a secure channel.
- $A \overset{o}{\rightarrow} S$: *Information* means that A sends *information* to S through an open channel.
- $A \overset{b}{\rightarrow} network$: *Information* represents when A broadcasts *information* to all on-line legitimate nodes.
- $A \overset{b}{\leftrightarrow} network$: *Information* represents a two-step procedure where A broadcasts *information* to all on-line legitimate nodes of the network, and receives their answers.
- h stands for a public hash function.
- T denotes the threshold period that a legitimate nodes may be off-line without beingn excluded of the network.

IV. GASMAN DESCRIPTION

This section contains the description of the procedures that form part of the GASMAN architecture, including all the specific details about network initialization, node insertion, access control, proofs of life, and node deletion.

A. Network Initialization

The proposed protocol requires the definition of an initialization phase where the secret information associated to the process of identification is generated and distributed within the initial network. This initialization phase consists in the definition of the graph used for the development of the protocol. Such a graph should be generated with the participation of all the original members of the network. Furthermore, the initialization phase also implies the distributed generation by the initial legitimate members of the network of a hard instance of the HCP in such a graph, task that was analyzed in [20].

In our proposal, as in trust graphs, the vertex set corresponds to the set of nodes in the actual network during its whole life-cycle. Consequently, the initialization process starts from a set V_0 of n vertexes corresponding to the nodes of the initial network. Hence, each vertex sub-index may be used as identification (ID) for the corresponding node. The first step of the initialization process consists of generating cooperatively and secretly a random permutation Π of such a set. Once this generation is completed, each legitimate node should know a Hamiltonian cycle HC_0 corresponding exactly to such a permutation. Finally, the partial graph formed by the edges corresponding to such a Hamiltonian cycle HC_0, is completed by adding n groups of

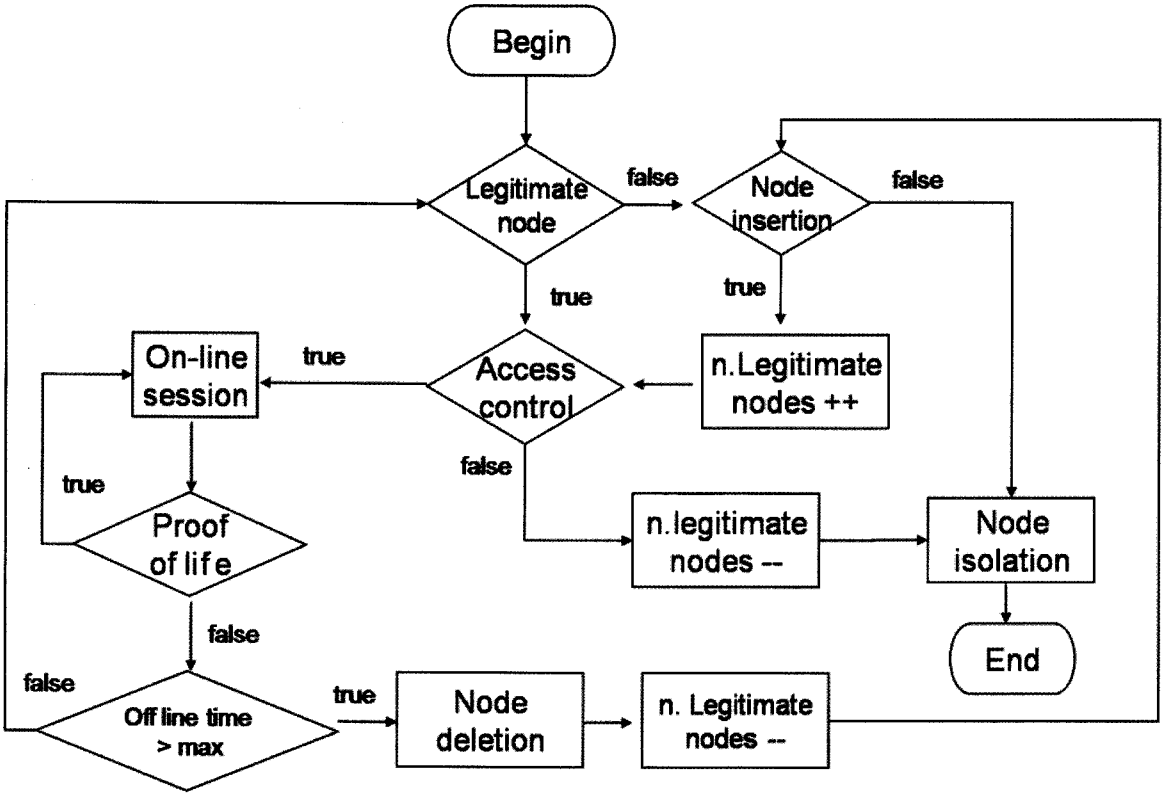

Fig. 1. Node life-cycle.

$2m/n$ edges, producing the initial edge set E_0. Here, m stands for the number of edges that the initial graph will have after the initialization stage. Each one of these n groups of edges will be generated by v_i, $i = 1, 2, \cdots, n$ according to the following restrictions: They must have v_i as one of its vertexes, while the other one will be randomly generated. Note that the size $2m/n$ of those edge subsets must be large enough so that the size of the resulting edge set $|E_0| = m$ guarantees the difficulty of the HCP in the graph G_0.

In general, finding Hamilton cycles is a difficult task even in relatively small graphs [21], [22]. However, since in our proposal it is necessary to guarantee the difficulty of the generated instance, we could use sparse pseudo-random regular graphs based on a generalization of knight's tours [23]. After the individual processes described in the previous paragraph, in order to generate cooperatively and secretly such a graph, the authenticated Diffie Hellmann key exchange protocols could be used [24].

Initialization algorithm

Input: V_0, with $|V_0| = n$.

1. The n nodes of the network generate cooperatively, secretly and randomly the cycle $HC_0 = \Pi(V_0)$.
2. $\forall v_i \in V_0$, v_i builds the set

$$N_{G_0}(i) = \{\{v_j \in_r V_0\} \cup N_{HC_0}(i)\}$$

with $|N_{G_0}(i)| = \frac{2m}{n}$.

3. $\forall v_i \in V_0 : v_i \xrightarrow{b} network : N_{G_0}(i)$.
4. $\forall v_i \in V_0 : v_i$ merges:

$$E_0 = \bigcup_{i=1,2,\cdots,n} \{(v_i, v_j) : v_j \in N_{G_0}(i)\}.$$

Output: $G_0 = (V_0, E_0)$, with $|E_0| = m$.

Once the creation of the initial instance of the problem has been carried out through the contribution of all the nodes of the network, each node will know a Hamiltonian cycle in the resulting $2m/n$-regular graph. From then on, each time a new user S wants to become a member of the network, it has to contact a legitimate member A in order to follow the insertion procedure explained in the following section.

B. Node Insertion

Let us suppose that we are at stage t of the network life-cycle when a user S contacts a legitimate member A of the network to become a member of the network. Once S has convinced A to accept its membership in, the first step that A should carry out is to assign S the lowest vertex number v_i not assigned so far in the vertex set V_t. Afterwards, A should broadcast such an assignment to all on-line legitimate nodes in order to prevent another simultaneous insertion with the same identifier. If A receives less than $n/2$ answers to the previous message, she stops the insertion procedure because the number of nodes that are aware of the insertion is not large enough. Otherwise, A develops the corresponding update of the secret Hamiltonian cycle HC_t by selecting at random two neighbor vertexes v_j and v_k in order to insert the new node v_i between them. Additionally, A chooses at random a subset of $2m/n - 2$ nodes in V_t such that none of

them is its neighbor in HC_t. Finally, A broadcasts the set of neighbors $N_{G_{t+1}}(v_i)$ of S in the new graph G_{t+1} .

Each time a node receives a graph update, it should secretly modify the corresponding Hamiltonian cycle. In order to achieve it, it uses the information provided to identify the unique position (according to the new edge set E_{t+1}) in the cycle where the new node can be inserted. In this way, it will be able to easily update the secret network key by simply inserting the vertex v_i between the vertexes v_j and v_k. At the same time, the authenticator node A must send the supplicant node S both the graph G_{t+1} (deploying an open channel), and the Hamiltonian cycle HC_{t+1} (through a secure channel).

`Insertion algorithm`

Input: At stage t a supplicant node S wants to become a member of the network.

1. $S \rightleftharpoons A$.
2. Node S convinces node A to accept its membership in the network.
3. A assigns S the identifier v_i such that $i = \min\{l : v_l \notin V_t\}$.
4. $A \overset{b}{\leftrightarrow} network : v_i$.
   - 4.1 If A receives less than $n/2$ answers, she stops the insertion procedure.
   - 4.2 Otherwise:
     - 4.2.1 A chooses: $(v_j, v_k) : v_j \in_r V_t, v_k \in_r N_{CH_t}(v_j)$.
     - 4.2.2 A chooses at random: $N_{G_{t+1}}(v_i) = \{v_j, v_k\} \cup \{w_1, w_2, \cdots, w_{\frac{2m}{n}-2}\}$ such that $N_{G_{t+1}}(v_i) \subseteq V_t \wedge \forall w_{l_1}, w_{l_2} : w_{l_1} \notin N_{CH_t}(w_{l_2})\}$.
     - 4.2.3 $A \overset{b}{\rightarrow} network : N_{G_{t+1}}(v_i)$.
     - 4.2.4 Each on-line node updates G_t by defining $V_{t+1} = V_t \cup \{v_i\}$, $E_{t+1} = E_t \cup N_{G_{t+1}}(v_i)$ and $HC_{t+1} = HC_t \setminus \{(v_j, v_k)\} \cup \{(v_j, v_i) \cup (v_i, v_k)\}$.
     - 4.2.5 $A \overset{o}{\rightarrow} v_i : G_{t+1}$.
     - 4.2.6 $A \overset{s}{\rightarrow} v_i : HC_{t+1}$.

Output: The supplicant node S becomes a legitimate member of the network.

C. Access Control

If a legitimate node S has been off-line or out-of-coverage from stage t and wants to re-enter into the network at stage r, its first step should be to contact a legitimate on-line member A. Afterwards, A should check whether the period S has been off-line is not greater than T. In this case, S has to be authenticated by A through a ZKP based on its knowledge of the secret solution HC_t on the graph G_t.

The aforementioned ZKP begins with the agreement between A and S on the number of iterations l to execute. From there on, in each iteration, S will choose a random permutation $\Pi_j(V_t)$ on the vertex set that will be used to build a graph $\Pi(G_t)$ isomorphic to G_t. The hash value of both the permutation $h(\Pi_j(V_t))$ and the Hamiltonian cycle in the graph $h(\Pi_j(HC_t))$ are then sent to A. When this information is received by A, it chooses a bit b_j at random ($b_j \in_r \{0, 1\}$). Depending on the selected value, S will provide A with the image of the Hamiltonian cycle through the isomorphism, or with the specific definition of the isomorphism. In the verification phase, A will check that the received information was correctly built.

Once the authentication of supplicant S has been successfully carried out, the authenticator A gives him the necessary information to have full access to the protected resources such as the chat application, for example.

`Access control algorithm`

Input: At stage r a supplicant node S that has been off-line since stage t wants to re-enter into the network.

1. $S \rightleftharpoons A$.
2. $S \overset{o}{\rightarrow} A : G_t$.
3. A checks whether $r - t \leq T$.
4. If $r - t > T$, then S is not authenticated.
5. Otherwise:
   - $A \leftrightarrow S : l$.
   - for $j = 1, 2, \cdots, l$.
   - 5.1 S chooses $\Pi_j(V_t)$ and builds $\Pi_j(G_t)$ and $\Pi_j(HC_t)$, the graph isomorphic to G_t and the corresponding Hamiltonian cycle, respectively.
   - 5.2 $S \overset{o}{\rightarrow} A : \{h(\Pi_j(V_t)), h(\Pi_j(HC_t))\}$.
   - 5.3 A chooses the challenge $b_j \in_r \{0, 1\}$.
   - 5.4 $A \overset{o}{\rightarrow} S : b_j$.
     - 5.4.1 If $b_j = 0$, then $S \overset{o}{\rightarrow} A : \{\Pi_j(G_t), \Pi_j(HC_t)\}$.
     - 5.4.2 If $b_j = 1$, then $S \overset{o}{\rightarrow} A : \Pi_j$.
   - 5.5 A verifies that
     - $\Pi_j(HC_t)$ is a valid Hamiltonian cycle in $\Pi_j(G_t)$, if $b_j = 0$.
     - the hash function h applied on $\Pi_j(G_t)$ coincides with $h(\Pi_j(G_t))$, if $b_j = 1$.
   - If $\exists j \in \{1, 2, \ldots, l\}$ such that the verification is negative, then S is isolated.
   - Otherwise $A \overset{s}{\rightarrow} S$: The necessary information to have full access to protected resources of the network.

Output: Node S is connected on-line to the network.

D. Proofs of Life

All on-line legitimate nodes have to confirm their presence in an active way. Such a confirmation is carried out every period of time T. It consists in broadcasting a message (proof-of-life) to all on-line legitimate nodes.

If some insertion happens during such a period, a proof of life of every on-line legitimate node will be distributed together with the information necessary for the insertion procedure. Otherwise, only the proof of life is required. During such a broadcast every node adds its own proof of life to the broadcast. In this way, when the broadcast reaches the last node, a broadcast back starts containing the proofs of life of all on-line legitimate nodes.

`Proof-of-life algorithm`

Input: At stage t node A is an on-line legitimate node of the network.

1. A initializes its $clock = 0$ just after its last proof of life.
2. If $clock > T$, then
   - 2.1 $A \overset{b}{\leftrightarrow} network : A's\ proof\ of\ life$.

2.1.1 If A receives less than $n/2$ proofs of life as answers to her broadcast, she stops her proof of life and puts back her clock.

2.1.2 Otherwise: $A \xrightarrow{b} network$: *Received proofs of life.*

Output: At stage $t+1$ node A continues being an on-line legitimate node of the network of the network.

Note that the possibility that a legitimate, but malicious, node can broadcast a fake proof of life for other nodes exists. However, the potential impact of this threat may be considered low since it would imply just the possible life extension of some off-line nodes.

E. Node Deletion

The deletion procedure is mainly based on the confirmation of the active presence of on-line legitimate nodes through their proofs of life. Each node should update its stored graph by deleting all those nodes that have not sent any proof of life after a period T. This fact implies that each node that has not proven its presence will be deleted from the network, as well as from the Hamiltonian cycle.

Node deletions are explicitly communicated to all on-line legitimate nodes in the second step of broadcasts of proofs of life. This way to proceed allows any node that is off-line in that moment will be able to update its stored graph as soon as it gets access to the network.

`Deletion algorithm`

Input: At stage t, a node v_i is an off-line legitimate node of the network.

1. A initializes her $clock = 0$.
2. If $clock > T$, then
   - 2.1 $\forall v_i \in V_t$: A checks v_i's proof of life in A's FIFO queue.
   - 2.2 A updates $V_{t+1} = V_t \setminus \{v_i \in V_t$ with no proof $\}$.
   - 2.3 A updates $E_{t+1} = E_t \setminus \{(v_i, v_j)$: $v_i \in V_t$ with no proof, $v_j \in N_{G_t(v_i)}\} \cup \{(v_j, v_k)$: $v_j, v_k \in N_{HC_t(v_i)}\}$.
   - 2.4 A updates $HC_{t+1} = HC_t \setminus \{(v_j, v_i), (v_i, v_k)\} \cup (v_j, v_k) : v_i \in V_t$ with no proof, $v_j, v_k \in N_{HC_t(v_i)}$.
3. If A started the broadcast used for the v_i's deletion, A adds this information to the second step of the proof-of-life broadcast: $A \xrightarrow{b} network : v_i$ is deleted.

Output: At stage $t+1$ the node v_i has been deleted both from the network and from the graph.

This procedure guarantees a limited growth of the graph that is used in authentication, and at the same time, allows that always the legitimate nodes set corresponds exactly to the vertexes in that graph. Apart from this, it is remarkable the fact that thanks to this procedure the recovery of legitimate members of the network that have been disconnected momentarily is possible.

V. ASSUMPTIONS AND SECURITY ANALYSIS

Note that the whole proposal is based on a single and shared secret network key and although the key is periodically updated, if a legitimate node is compromised and reveals the shared secret key, the whole network would be compromised [25], [26]. Consequently, this proposal initially assumes the ideal environment where all legitimate nodes are honest and where no adversary may compromise a legitimate node of the network in order to read its secret stored information. Such assumptions are well suited as a basic model in order to decide under which circumstances the GASMAN is applicable to MANETs. For instance, a possible adaptation of the proposal in order to avoid those hypothesis could be defining a threshold scheme to be used in every step of the GASMAN, so that every proof of life, insertion, access control or deletion operation should be done by a coalition of on-line nodes. Then, a dishonest node would not affect the correct operation of the network.

It is clear that the proposal inherits some problems of the distributed trust model such as the important necessity that legitimate nodes cooperate. Consequently, it is advisable to include a scheme to stimulate node cooperation.

Finally, another requirement of the GASMAN is the establishment of a secure channel for the insertion procedure. However, that aspect may be easily fulfilled thanks to the fact that most wireless devices are able to communicate with each other via Bluetooth wireless technology or through other more secure short range wireless methods.

With respect to possible attacks and due to the lack of a centralized structure, it is natural that possible DoS attacks have as their main objective the chat application. In order to protect the GASMAN against this threat it must be assured that chat messages, although are publicly readable, may be only sent by legitimate on-line members of the network. Another important aspect related to the use of the chat application is the necessary synchronization among the on-line nodes. In order to achieve it, we could use global time synchronization derived from the application of IEEE 802.11 timer synchronization function to MANETs [27].

MANETs are especially vulnerable to different threats such as identity theft (spoofing) and the man-in-the-middle attack. Such attacks are difficult to prevent in environments where membership and network structure are dynamic and the presence of central directories cannot be assumed. However, our proposal is resistant to spoofing attacks because access control is granted through a ZKP. It implies that any information published through the chat application or sent openly during the execution of access control mechanism becomes useless.

On the other hand, the goal of the man-in-the-middle attack is either to change a sent message or to gain some useful information by one of the intermediate nodes. Again, the use of ZKPs in our protocol implies that reading any transferred information does not reveal any useful information about the secret, so changing the message is not possible since only legitimate nodes whose access has been allowed can use the chat application.

Another active attack that might be especially dangerous in MANETs is the so-called Sybil attack. It happens when a node tries to get and use multiple identities. The most extreme case of this type of attacks is the establishment of a false centralized authority who states the identities of legitimate members. However, this specific attack is not possible against our scheme due

to its distributed nature. In the GASMAN, the responsibility of controlling general Sybil attacks will be shared among all the on-line nodes. If an authenticator node detects that a supplicant node is trying to get access to the network by using an ID that is yet being used on-line, such access control must be denied and the corresponding node must be isolated. The same happens when any on-line node detects that an authenticator node is trying to insert a new member into the network with a new ID, and such a node has yet assigned as a vertex ID. Again, such insertion must be denied and the corresponding supplicant node must be isolated. Anyway, if a Sybil attacker enters the network, any of its neighbors will detect it as soon as it sends proofs of life for different vertexes ID.

Finally, in the proposal, an eavesdropping node could observe all the exchanged messages and the zero-knowledge property guarantees that no important information about the shared secret is revealed. With respect to a possible play-back attack, by using the access control of our protocol, the on-line node A always can choose any random challenge, and the supplicant node S has to compute the correct response, which is later used by A to check if the authentication is successful. Therefore, previously used challenges and answers are useless.

VI. PERFORMANCE ANALYSIS

We now analyze the efficiency of the proposal both from the energy consumption and from computational complexity points of view. We consider the energy consumption which is the result of transmissions of data and processor activities due to authentication tasks. In the proposal there are two phases when computational overhead is more significant: The ZKP-based access control and the periodic checking of stored elements in the FIFO queue. A reduction on the number of rounds of ZKP has a direct effect on the total exchanged messages size in insertions, but a trade-off should be maintained between protocols robustness and performance. Indeed, regarding total data transmission over wireless links, the ZKPs take less than 10% in a usual situation.

The dominant time-consuming jobs are the periodic proofs of life, which accounts for around 90% of the total exchanged message size in many cases. However, we found that these compulsory proofs of life imply an incentive technique for stimulating cooperation in authentication tasks. This is due to the fact that nodes that are broadcasters of deletion queries or authenticators in insertions or access controls are exempted from their obligation to broadcast their proofs of life.

In order to reduce data communication cost, an increase on the threshold period T might be an option, but again an acceptable balance should be kept because T has implications also on storage requirements of the protocol. According to our experiments, T should depend directly on the number of legitimate and/or on-line nodes in order to prevent a possible bandwidth overhead in large networks.

For the performance analysis of the proposal, we used the network simulator NS-2 with the DSR routing protocol. We created several tcl based NS-2 scripts in order to produce various output trace files that have been used both to do data processing and to visualize the simulation. Within our simulation, we used the visualization tool of network animator (NAM) and the NS-2 trace files analyzer of tracegraph. For the simulation of mobility, we used the setdest program in order to generate movement pattern files using the random waypoint algorithm.

Table 1. Example of trace.

Time	Event	HC
0.1	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 are legitimate	8,3,9,7,4,2,6,5,1,10,0
1.2	Insertion of Node 14 is broadcast by Node 4	8,· · ·,4,14,2,· · ·,0
1.3	Nodes 3, 1, 0 do not answer to proof of life	
3.2	Node 0 is re-inserted by ZKP with Node 8	
8.6	Node 3 is re-inserted by ZKP with Node 4	
9.4	Node 1 is re-inserted by ZKP with Node 10	
11.6	Node 1 turns off	
13.9	Proof of life started by Node 3	
14.2	Nodes 1, 2 do not answer to proof of life	
14.8	Node 2 is re-inserted by ZKP with Node 14	
17.2	Proof of life started by Node 2	
17.5	Nodes 1, 5 do not answer to proof of life	
21.7	Node 5 turns off	
31.4	Node 1 turns on and Node 2 is chosen for ZKP	
31.5	Node 4 turns off	
32.5	Proof of life started by Node 1	
32.8	Nodes 4, 5, 6 do not answer to the proof of life	
34.2	Node 6 is re-inserted by ZKP with Node 2	
38.5	Proof of life started by Node 6	
38.7	Nodes 4, 5 do not answer to proof of life	
41.4	Node 1 turns off	
53.2	Node 1 turns on and Node 0 is chosen for ZKP	
59.6	Proof of life started by Node 6	
59.9	Nodes 4, 5 do not answer to proof of life	
64.2	Node 5 is deleted	8,· · ·,6,1,10,0
64.7	Node 2 turns off	
72.5	Node 4 turns on and Node 0 is chosen for ZKP	
75.3	Insertion of Node 13 is broadcast by Node 14	8,· · ·,2,13,6,1,10,0
75.4	Node 2 does not answer to proof of life	

An excerpt of the trace files corresponding to the an example of simulation is shown in Table 1. Basically, it consisted of generating a scenario file that describes the movement pattern of the nodes and a communication file that describes the traffic in the network. These files were used to produce trace files that were analyzed to measure various parameters.

The trace files are used to visualize the simulation using NAM, while the measurement values are used as data for plots with tracegraph. The final graph and Hamiltonian cycle associated to the example network is shown in Fig. 2, where the Hamiltonian cycle, the inserted nodes and the edges deleted from the Hamiltonian cycle when inserting new nodes are shown.

In order to study the effectiveness of the GASMAN, we studied it in a set of realistic scenarios. In particular, we used the most commonly used mobility model by the research community, the so-called random waypoint model, which uses pause times and random changes in destination and speed.

An extensive number of simulations using NS-2 simulator with 802.11 MAC and DSR routing protocols in order to see the effects of different metrics by varying network density and

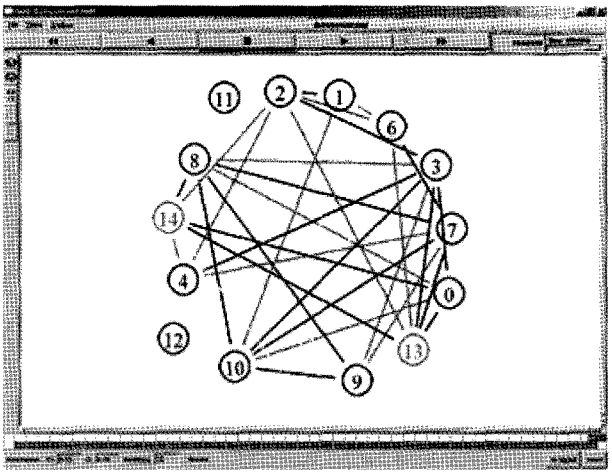

Fig. 2. Example of final associated graph and Hamiltonian cycle.

topology were run. Within the simulations, relationships can be established anytime two nodes are located in close proximity and the random walk mobility model was used with various pause time and maximum speed. In particular, we varied the number of nodes from 15 to 100. Also, our architecture was evaluated with 250×250, 500×500, and 750×750 m^2 square area of ad-hoc network. In each case, the nodes move around with 0.5 second pause time and 20 m/s maximum speeds. The transmission range of the secure channel is 5 meters while that of the data channel is fixed to 250 meters. The period of simulation varied from 60 to 200 seconds. We also changed the probabilities of insertions and deletions in each second from 5% to 25%, in order to modify the mobility rate and antenna range of nodes from 2 to 15 m/s and 100 to 250 m, respectively. This range also defines different frequencies of accesses to the network.

The first conclusions we obtained from the simulations are:

- The protocol scales perfectly to any sort of networks with different levels of topology changes.
- Node density is a key factor for the mean time of insertions, but such a factor is not as big as it might be previously assumed since nodes do not forward two packets of data corresponding to the same proof of life coming from two different nodes.
- A right choice of parameter T should be done according to number of nodes, bandwidth of wireless connections and computation and storing capacities of nodes.
- A positive aspect of the proposal is that the requirements in the devices' hardware are very low.

VII. CONCLUSIONS AND OPEN QUESTIONS

Successful authentication in mobile ad-hoc networks is critical for assuring secure and effective operation of the supported application. This work describes a new authentication scheme, the so-called GASMAN, which has been specially designed for MANETs. Such a protocol supports knowledge-based member authentication in server-less environments. The overall goal of the GASMAN has been the design of a strong authentication scheme that is able to react and adapt to network topology changes without the necessity of any centralized authority. In order to avoid the transference of any relevant information, its core technique consists of a zero-knowledge proof. Furthermore, the proposal is balanced since the procedures that the legitimate members of the network have to carry out when the network is updated (insertion or deletion of nodes) imply identical work for every legitimate member of the network.

The development of an initial simulation of the proposal through the NS-2 network simulator has been carried out. A statistical analysis of the proposal and a comparison of simulation results with other approaches will be included in a forthcoming version of this work. Finally, two important tasks included among future works are the improvement of formal description and verification of the proposal by using the BAN logic, and the implementation of the proposal on real devices to get the realistic processing performance.

REFERENCES

[1] N. Aboudagga, M. Tamer, M. Eltoweissy, L. DaSilva, and J. J. Quisquater, "Authentication protocols for ad-hoc networks: Taxonomy and research issues," in *Proc. 1st ACM international workshop on Quality of service and security in wireless and mobile networks,* Oct. 2005.

[2] A. Weimerskirch, "Authentication in ad-hoc and sensor networks," Ph.D. thesis, Ruhr-University Bochum, Germany, July 2004.

[3] L. Zhou and Z. Haas, "Securing ad-hoc networks," *IEEE Network*, vol. 13, pp. 24–30, 1999.

[4] H. Luo and S. Lu, "Ubiquitous and robust authentication services for ad hoc wireless networks," Department of Computer Science, UCLA, Tech. Rep. TR-200030, 2000.

[5] J. Kong, H. Luo, K. Xu, D. L. Gu, M. Gerla, and S. Lu, "Adaptive security for multi-level ad-hoc networks," *J. Wireless Commun. Mobile Comput.*, pp. 533–547, 2002.

[6] S. Jarecki, N. Saxena, and J. H. Yi, "An attack on the proactive RSA signature scheme in the URSA ad hoc network access control protocol," in *Proc. ACM Workshop on Security of Ad Hoc and Sensor Networks,*, 2004, pp. 1–9.

[7] J. P. Hubaux, L. Buttyán, and S. Capkun, "The quest for security in mobile ad-hoc networks," in *Proc. MobiHoc*, 2001, pp. 146–155.

[8] Y. Kim, D. Mazzocchi, and G. Tsudik, "Admission control in peer groups," in *Proc. IEEE International Symposium on Network Computing and Applications*, 2003.

[9] S. Hahm, Y. Jung, S. Yi, Y. Song, I. Chong, and K. Lim, "A self-organized architecture in mobile ad-hoc networks," in *Proc. ICOIN*, 2005, pp. 689–696.

[10] N. Saxena, G. Tsudik, and J. H. Yi, "Efficient node admission for short-lived mobile ad-hoc networks," in *Proc. ICNP*, Nov. 2005, pp. 269–278.

[11] O. Goldreich, S. Micali, and A. Wigderson, "How to prove all NP-statements in zero-knowledge, and a methodology of cryptographic protocol design," in *Proc. Crypto*, vol. 263, 1986, pp. 171–185.

[12] H. Asaeda, M. Rahman, H. Manshaei, and Y. Fukuzawa, "Implementation of group member authentication protocol in mobile ad-hoc networks," in *Proc. WCNC*, Las Vegas, USA Apr. 2006.

[13] A. Wierzbicki, A. Zwierko, and Z. Kotulski, "A new authentication protocol for revocable anonymity in ad-hoc networks," in *Proc. CNIS*, 2005.

[14] P. Caballero-Gil and C. Hernández-Goya, "Zero-knowledge hierarchical authentication in MANETs," *IEICE Trans. Inf. Syst. Lett.*, E-89-D, pp. 1288–1289, 2006.

[15] P. Caballero-Gil and C. Caballero-Gil, "A global authentication scheme for mobile ad-hoc networks," in *Proc. IWSEC*, 2007, pp. 105–120.

[16] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, and A. Quesada-Arencibia, "A simulation study of new security schemes in mobile ad-hoc networks," in *Proc. EUROCAST*, 2007, pp. 73–81.

[17] S. Maki, T. Aura, and M. Hietalathi, "Robust membership management for ad-hoc groups," in *Proc. NORDSEC*, 2000.

[18] D. Glynos, P. Kotzanikolaou, and C. Douligeris, "Preventing impersonation attacks in MANET with multi-factor authentication," in *Proc. WIOPT*, 2005, pp. 59–64.

[19] P. Caballero-Gil and C. Hernández-Goya, "Strong solutions to the identification problem," in *Proc. COCOON*, 2001, pp. 257–261.

[20] B. Krishnamachari, S. Wicker, R. Bejar, and C. Fernández, "On the complexity of distributed self-configuration in wireless networks," *Telecommunication Systems*, vol. 22, pp. 33–59, 2002.

[21] B. Vandegriend, "Finding hamiltonian cycles: Algorithms, graphs and performance," M.S. thesis, University of Alberta, Canada, 1998.
[22] I. B. Shields, "Hamilton cycle heuristics in hard graphs," Ph.D. thesis, North Carolina State University, 2004.
[23] M. Krivelevich and B. Sudakov, "Sparse pseudo-random graphs are Hamiltonian," *J. Graph Theory*, vol. 42, no. 1, pp. 17–33, 2002.
[24] W. Diffie, P. C. van Oorschot, and M. J. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and Cryptography*, pp. 107–125, 1992.
[25] Y. Hao, L. Haiyun, Y. Fan, L. Songwu, and Z. Lixia, "Security in mobile ad-hoc networks: Challenges and solutions," *Wireless Commun.* vol. 11, no. 1, pp. 38–47, 2004.
[26] T. Gene, "Some issues in WSN, MANET, and cellular security," in *Proc. ARO Planning Workshop on Embedded Systems and Network Security,* 2007, pp. 22-23.
[27] X. Zhao, V. Ganapathy, N. Pissinou, and K. Makki, "Revisiting global time synchronization," in *Proc. IEEE GLOBECOM*, 2007, pp. 1058–1063.

Pino Caballero-Gil was born in San Bartolomé de Tirajana, Las Palmas de Gran Canaria, Spain, on November 29, 1968. She received the M.S. and the Ph.D. degrees in Mathematics from the University of La Laguna, Spain in 1990 and 1995, respectively. She is Associate Professor at the University of La Laguna since 1997 and the Coordinator of the Research Group CryptUll at the University of La Laguna since 1990. Currently, she is the Dean of the Faculty of Mathematics. Her major interests are security in vehicular ad hoc networks, stream ciphers, authentication, and cryptographic protocols.

Candelaria Hernández-Goya was born in Santa Cruz de Tenerife, Spain, on June 18, 1970. She received the M.S. and the Ph.D. degrees in Mathematics from the University of La Laguna, Spain in 1995 and 2003, respectively. She is Lecturer at the University of La Laguna since 1998. Her major interests are security in vehicular ad hoc networks, authentication, and cryptographic protocols.