An Efficient Group Key Management for Secure Routing in Ad Hoc Networks

Natalia Castro Fernandes and Otto Carlos Muniz Bandeira Duarte
GTA/PEE/COPPE - Universidade Federal do Rio de Janeiro
Rio de Janeiro, Brazil

***Abstract*—This paper proposes and specifies a protocol for distributing and managing group keys in ad hoc environments, which applies for the Secure Optimized Link State Routing Protocol. Our protocol manages group keys taking into consideration the frequent network partitions and the absence of infrastructure. The analysis shows that the protocol is energy efficient for high key replacement rates and frequent network partitions. The proposal reduced up to 512 times the control traffic load and 356 times the energy spent with cryptographic operations when compared to contributory algorithms. The proposed protocol is robust even in the presence of non-cooperative nodes and provides an efficient key management in a timely manner.**

I. Introduction

Mobile Ad Hoc Networks (MANET) technology allows a self-organized wireless connection of mobile devices. The provision of security to ad hoc networks, however, faces many specific vulnerabilities. First, the wireless link is vulnerable to passive and to active attacks, like eavesdropping and jamming, respectively. Moreover, ad hoc networks are based on collaborative routing, which means that a node working in a malicious way may disrupt the entire network. Therefore, many secure routing protocols were proposed, but they rely on key management systems, which are still an open problem.

Key management is a challenge in ad hoc networks because it is not possible to guarantee the availability of a resource to all nodes at any time. Hence, ad hoc networks cannot base its authentication system in centralized and fixed infrastructure. Furthermore, ad hoc networks usually are composed by nodes with constrained devices. Hence, security must be provided without large energy consumption, because some nodes cannot execute frequent complex cryptographic operations.

In this paper we propose a protocol to manage group keys in ad hoc environments which use Secure Optimized Link State Routing Protocol (SOLSR) [1]. The proposed protocol, called Efficient Group key management for Secure Routing (EGSR), uses a small number of messages in the group key distribution process to reduce energy consumption. EGSR is composed of three main mechanisms: group key distribution; fusion of network partitions and new nodes joining; and round leader failure detection and replacement. In the protocol, the group key is periodically replaced to exclude non-authorized nodes and to avoid the use of the same group key in more than some amount of data, especially when weak encryptions techniques are in use. Our proposal is compatible with ad hoc characteristics, such as the absence of infrastructure and the frequent network partitions. EGSR does not need an initialization phase in which the administrator prepares some nodes with secrets which can disrupt the entire network if exposed. In our protocol, all nodes only need a public and a private key, a certificate given by the access control entity and to be on the authorized node list to start using the network. The proposed protocol simplifies the exclusion of non-authorized nodes and the detection of bad behavior.

The remainder of the paper is structured as follows. In Section II, we discuss related work. In Section III, we describe the system model. In Section IV, we show the details of the proposed protocol and, in Section V, we show the analysis results. In Section VI, we present the conclusions.

II. Related Work

One cryptographic scheme suitable for ad hoc networks to establish group keys is the contributory key agreement. In these protocols, all nodes cooperate to form a new group key. This approach is completely distributed and reduces the chances of choosing a weak group key [2], [3], [4]. Nevertheless, these protocols overcharge network with the messages to form a new key.

Key pre-distribution schemes address the key distribution in networks composed of constrained devices. In this approach, an administrator selects a pool of keys from the key space. Each node receives a random subset from the key pool before network deployment. Any pair of nodes with a common key within their subsets can use that key to establish a secure communication. After the stabilization of secure links, nodes can choose a group key [5], [6].

Cluster based protocols aim to build a scalable key management [3], [7]. One approach to distribute group keys on multicast environments based on clusters is the Optimized Multicast Cluster Tree with Multipoint Relays (OMCT with MPR), whose main idea is to use information of Optimized Link State Routing Protocol (OLSR) [8] to elect the local controllers of the created clusters [9]. OMCT with MPRs assumes that routing control messages have been exchanged before the key distribution. In SOLSR, however, all routing control messages must be signed. Therefore, key distribution must be deployed before the exchange of routing control messages. Then, OMCT with MPRs is not useful to distribute a group key to SOLSR.

Our proposal for managing group key in ad hoc networks, EGSR, avoids message overhead, differently from contributory key agreement protocols. Besides, our protocol does

not depend on the establishment of secrets before network deployment [10], [5]. Therefore, even if authorized nodes are hacked, network security is not completely compromised.

III. System Model

A. Network Model

Our protocol works under the assumption of mobile nodes which collaboratively support network operation. Network partitions can occur at any time and membership can change frequently. We define as group the set of nodes which can communicate through routes of one or more hops. Nodes in the same group must share the same group key to exchange routing control messages. We suppose that nodes run Secure Optimized Link State Routing Protocol (SOLSR), a link state proactive routing protocol. Then, all nodes always know the number of nodes with which they can communicate. Besides, SOLSR controls flooding with a mechanism called Multipoint Relays (MPRs). In this mechanism, only nodes selected as MPR forward control messages. MPR nodes are selected by each node amongst the set of one hop neighbors, in a way to reach all two-hop neighbors. Also, nodes discover the approximately delay between its clocks in SOLSR to avoid replay attacks. This information is used in our proposal to establish a weak synchronization on the network.

We assume that an access control entity (ACE) controls network membership [10], [11]. The ACE sends an alert when nodes join or leave the network and creates certificates to each node, associating a public key K_i to some identity ID_i. In our model, the membership of a group is formed by the authorized nodes, which use network resources in a legitimate way. For a node to become authorized, the access control entity must announce the node identification ID_i to the network. Besides, an authorized node must have a public key K_i, a private key k_i and the certificate C_i signed by the access control entity to obtain the group key. Each authorized node is also responsible for maintaining a list of all authorized nodes. This is not a great issue because we assume the use of SOLSR, which maintains a list of all the active nodes in the network.

IV. The proposed Scheme

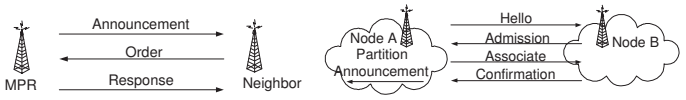
A. Overview

The proposed protocol, EGSR, distributes the group key to all nodes using asymmetric cryptography based on three main mechanisms: the group key distribution, for establishing the group keys; the new nodes joining, partition fusion and network initialization; and the leader failure detection and leader replacement. In EGSR, the group key distribution is initialized in each round by a round leader, and if the round leader fails, it is necessary to automatically substitute the round leader to continue the group key distribution.

B. Group Key Distribution

The group key distribution mechanism replaces the group key periodically or when a node is excluded. The periodic distribution excludes adversaries which possess the group key, but not a private key. For instance, in community networks, an authorized user may send the group key to a non-authorized friend in order to the friend accesses network resources. The group key distribution is also triggered by an intrusion detection system (IDS). When the IDS sends an alert, it means that there is an adversary that should be excluded.

Figure 1(a) illustrates the group key distribution mechanism. The round leader initiates the group key distribution through the broadcast of an Announcement message, which indicates the existence of a new group key. When the neighbors of the round leader listen to the Announcement, they send the Order message to receive the new group key. The round leader ends the process with each neighbor sending the Response message, which contains the new group key encrypted with its neighbor's public key. The neighbors that are leader's multipoint relays (MPRs) further retransmit the Announcement, and neighbors by two hops choose an MPR to obtain the new group key. All MPRs of the network repeat this mechanism to guarantee that all nodes will receive the new group key.



(a) Group key distribution. (b) Partition fusion.

Fig. 1. Mechanisms of EGSR.

The messages are on Figure 2(a), 2(b) and 2(c). The fields for signature, certificate, and encrypted key have variable size, depending on the hash function, cryptography algorithms, and key size. The certificate and the message signature are important to authenticate the node sending the message and to guarantee the content integrity. The key distribution for each pair of nodes is successful only if both nodes prove that their identities are on the authorized node list. The other message fields identify the parameters of the next key distribution.

Nodes must begin to use the new group key approximately at the same time. Therefore, each node calculates the expected time to start using the new group key, T_w, given by

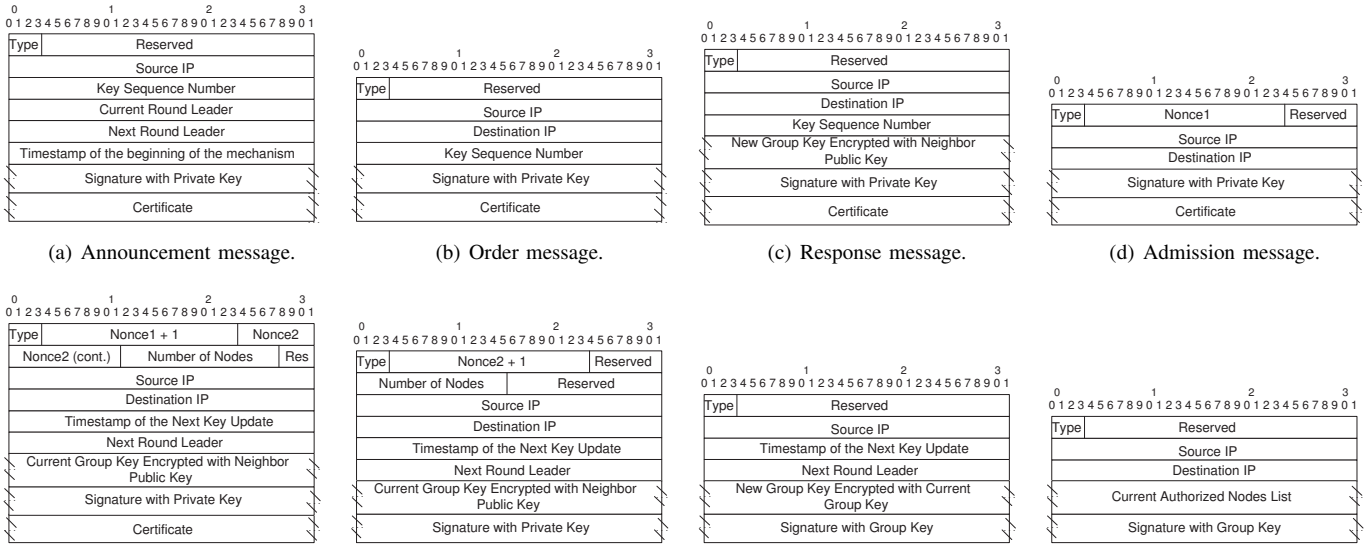
$$T_w = T_b + T_n * H_{max}. \quad (1)$$

In this equation, T_b is the approximate time when the group key distribution began, which is obtained in the Announcement message, T_n represents an estimative of the maximum delay a MPR takes to transmit the new group key to its neighbors and H_{max} represents the number of hops between the round leader and the node more distant from it in the network.

Nodes start to use the new group key after T_w, although they accept messages signed with old or new group key in the period given by $T_w - \alpha$ and $T_w + \alpha$, where α represents the delay tolerance. After $T_w + \alpha$, messages not signed with the new group key are discarded. Nodes that did not receive the group key before $T_w + \alpha$ are treated as new nodes, and they obtain the group key with the joining mechanism, described in section IV-C. Due to α and to the node joining mechanism, EGSR needs only a weak synchronization.

C. Joining Nodes, Partition Fusions and Initialization

Group key distribution mechanism treats excluded nodes, but not joining nodes. When an authorized node joins the network, it must obtain the current group key. Similarly,



(a) Announcement message. (b) Order message. (c) Response message. (d) Admission message.

(e) Associate message. (f) Confirmation message. (g) Partition Announcement message. (h) List message.

Fig. 2. EGSR messages.

when two network partitions restore a common link, they must establish a common key, so that all the routing control messages are accepted by the nodes of both partitions.

Figure 1(b) shows the partition fusion mechanism. When node B receives a HELLO message from an authorized node, such as node A, signed with a different group key, it unicasts the Admission message to signal the detection of the partition. When node A receives the Admission message, it verifies if node B is on the authorized node list and absent of the active nodes list of SOLSR. Then, node A answers B with an Associate message, informing the current parameters of A's partition, like the group key and the number of nodes in the partition. After receiving Associate, B sends the Confirmation message, indicating the key reception and sending information of B's partition. The Admission, Associate, and Confirmation messages are on Figures 2(d), 2(e), and 2(f).

After the key exchange, the partitions must share the same group key. Hence, the node in the smallest partition announces itself as immediate round leader and distributes the new group key, flooding the Partition Announcement message (Figure 2(g)). In the partition fusion, we do not authenticate the nodes, because all nodes with the group key are trusted.

The joining nodes mechanism is similar to the partition fusion mechanism. Admission, Associate and Confirmation messages are exchanged among the nodes. Then, the List message, represented in Figure 2(h), is transmitted to the new node after the reception of the Confirmation message to inform the current authorized nodes list. Another similar mechanism is the network initialization, which consists of many partition fusion/joining node mechanisms. Therefore, nodes form groups with one hop neighbors and these small groups will further fuse like if each group was a partition. To avoid loops in this mechanism, we use a decision process based on two rules. The first rule is that, if there is more than one partition fusion at the same time, the Partition Announcement message of the smallest partition are always discarded and the group key of the partition with more nodes is adopted. The second rule is that, if both partitions have the same size, the partition with the leader with the greatest IP will predominate and the message of the other partitions will be discarded. Therefore, all nodes will obtain the same group key after all the network partition fusions.

D. Round Leader Failure Detection

The round leader chooses the group key and starts the key distribution. The round leader selection follows a rule to stop colluding malicious nodes from being always the round leaders and choosing weak group keys. Each round leader selects the next round leader by ordering the IPs of the active nodes and selecting the node after itself on the list. Nevertheless, the round leader is still a failure point of the key distribution. If a round leader fails on starting the group key distribution, the group key management would be compromised. Hence, we use a mechanism to detect round leader failure and to replace the leader. Nodes detect a round leader failure when a group key distribution is pretended to start, but no neighbors sent the Announcement Message after T_k, given by

$$T_k = T_{b_{new}} + T_n * N_h + \delta, \tag{2}$$

where N_h is the number of hops from the round leader up to the node, and δ is the delay tolerance. The variable T_n is an estimate of the maximum delay with group key distribution from an MPR to its neighbors, and $T_{b_{new}}$ is the expected time to the key distribution begin. $T_{b_{new}}$ is achieved by summing the interval between automatic key replacements and the timestamp of the beginning of the last group key distribution mechanism. The round leader is considered absent if the new key is not received up to T_k.

When a node detects a round leader failure, it selects the next round leader by putting in order the IPs of the active nodes and choosing the next node after the current round leader. Every node calculates a new T_k, considering the delay until the new round leader detects the current round leader

absence. The time to start using the new group key, T_w (Equation 1), is restarted to the new round leader. The round leader replacement mechanism is accomplished for a node when it obtains the new group key. If it obtains different group keys with delays smaller than T_w, but greater than T_k, the node accepts the oldest round leader key and updates its T_n.

V. Protocol Analysis

A. Petri Network Analysis

The state machine of EGSR (Figure 3) was converted into a Petri network to evaluate the protocol. The tool ARP, version 2.3 [12] was used in the analysis. The results show that the protocol has the expected properties [13]: boundedness, because protocol has a finite number of states; liveness, as there are no dead-locks, representing that all actions of the protocol are possible; and repetitiveness, because it is possible to return to initial state from any state in network. Then, our protocol can be implemented and has no loops or states from where is not possible to reach some other state.

B. Security Analysis

In this section, we discuss how our protocol and an intrusion detection system (IDS) handle potential security issues.

1) Group Key Disclosure: If a non-authorized node obtains the current group key G_n, it can sign routing control messages. Assume that S_{ID} is the set of the identifications of all authorized nodes. If the non-authorized node chooses a randomly identification ID_k, $ID_k \notin S_{ID}$, this node will not access any resource of the network, because all nodes know S_{ID} from the authorized nodes list. On the other hand, if the non-authorized node knows an identification of an absent authorized node ID_m, $ID_m \in S_{ID}$, authorized nodes cannot immediately identify the intrusion. Suppose that f_r is the frequency of the automatic group key distribution, which replaces the group key. Non-authorized node will stop using the network in a period $p \leq 1/f_r$, because the non-authorized node does not have the private key k_m and the certificate C_m required by the Order message in the group key distribution. Besides, if the non-authorized node do malicious actions, the IDS can detect and send an alarm before the next automatic group key distribution. This alarm triggers a group key distribution, excluding the non-authorized node quickly.

2) Private Key Disclosure: A worse case than group key disclosure is the private key disclosure. In this case, non-authorized node has not only the current group key G_n, but also an ID_m, $ID_m \in S_{ID}$, the private key k_m, the public key K_m, and the certificate C_m of an authorized node. With this material, non-authorized node can sign any control message and authenticate itself in the group key distribution. However, if $S_{ID_A} \subset S_{ID}$ is the set of active authorized nodes and $ID_m \in S_{ID_A}$, there will be at least two authentications of ID_m in the group key distribution, which indicates a malicious action that can be detected by the IDS. Then, ID_m is blocked and a new group key distribution process is started. If $ID_m \notin S_{ID_A}$, but $ID_m \in S_{ID}$, which means that the non-authorized node has an identity of an absent authorized node, the non-authorized node cannot be excluded based only in EGSR until the return of the authorized node.

C. Performance Analysis

We analyzed the energy consumption of EGSR and of group key agreement protocols with Matlab 7, considering energy constrained devices. The energy costs with cryptographic operations are relative to "StrongARM" microprocessor, designed for embedded low-power environments [3], [14]. We choose RSA with 1024-bit key, Advanced Encryption Standard (AES) with 128-bit key, and keyed-Hash Message Authentication Code (HMAC) with 128-bit key as cryptography functions, because they are well-known and largely used. We estimate the average number of messages sent and received by each node and the number of cryptographic operations carried out. Our scenario, which is denser than one of a community network [15], is composed of 256 nodes. We consider that the average number of neighbors of each node is approximately constant even with the mobility.

Figure 4 depicts the sum of the energy costs of all nodes with cryptography to distribute the group key in the group key distribution mechanism ($EGSR_{DIST}$), the partition fusion mechanism ($EGSR_{PART}$) and the worst case of the initialization phase ($EGSR_{INIT}$) of EGSR. It also shows the sum of energy costs of all nodes with cryptography using Burmester-Desmedt (BD) [3] and Group Diffie-Hellman (GDH.3) [2], two group key agreement algorithms. Both algorithms are based on the generalization of the Diffie-Hellman problem to a group. In BD, all nodes spend the same amount of energy, but, in GDH.3, there is a special node responsible for executing more cryptographic operations. Both BD and GDH.3 assume that all group members can hear any message. In our analysis, network routes are not established yet, so all messages of these protocols are flooded. The analysis of Figure 4 discards energy consumed with the authentication in EGRS because BD and GDH.3 only treat the cryptographic operations to obtain a new group key. Therefore, we just compare the energy on the key distribution/agreement. We observe that all EGSR mechanisms are less expensive than BD and GDH.3. The BD protocol consumes up to 1.6 times more energy than EGSR initialization, 1.7 times more energy than the group key distribution of EGSR, and 356 times more energy than the partition fusion of EGSR.

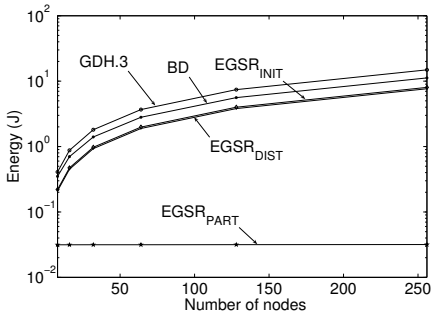


Fig. 4. Energy spent with cryptographic operations.

We also analyzed the impact with message transmission or retransmission in all of the protocols. The result is on Figure 5. BD and GDH.3 have the worst result because they flood all of the control messages. EGSR initialization showed

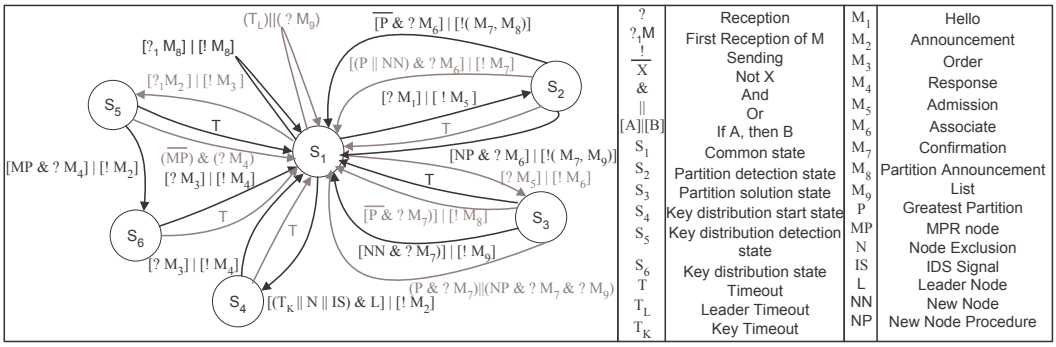


Fig. 3. EGSR state machine.

again the worst result among EGSR mechanisms, because an initialization phase corresponds to many partition fusion mechanisms. Although the partition fusion mechanism has the smallest consumption with cryptographic operations, it has approximately the same behavior than the group key distribution mechanism in terms of number of transmissions, once both mechanisms "flood" an information. The BD and GDH.3 protocols transmit up to 137 times more messages than EGSR initialization, and 512 times more messages than the group key distribution and the partition fusion of EGSR.

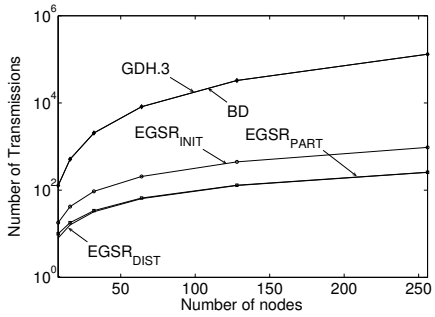


Fig. 5. Number of transmissions/retransmissions.

VI. Conclusions

In this paper, we presented and evaluated the Efficient Group key management for Secure Routing protocol (EGSR). Our protocol restricts non-authorized access to the network through periodic and triggered group key replacement. EGSR with SOLSR makes ad hoc routing more secure against non-authorized nodes with a small energy cost, even if there is collusion between authorized and non-authorized nodes. Moreover, the proposed protocol synchronizes the new group key use and is robust against node failures and network partitions. The use of an intrusion detection system increases the security provided by EGSR, because non-authorized nodes that utilize the private key of some authorized node to obtain the new group key are also excluded from network.

The analysis shows that EGSR works correctly and is implementable. Besides, it is adequate to energy constrained devices and simplifies the non-authorized node detection and exclusion on environments in which security is based on symmetric group keys. The analysis showed that EGSR consumes less energy and transmits fewer messages than BD and GDH.3, which are known protocols of group key agreement. Therefore, the use of EGSR turns routing in ad hoc networks more secure and does not significantly impact network performance.

References

[1] A. Hafslund, A. Tonnesen, R. B. Rotvik, J. Andersson, and O. Kure, "Secure extension to the OLSR protocol," in *OLSR Interop and Workshop*, San Diego, California, Aug. 2004, pp. 1–4.

[2] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, pp. 769–780, Aug. 2000.

[3] J. C. M. Teo and C. H. Tan, "Energy-efficient and scalable group key agreement for large ad hoc networks," in *2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks (PE-WASUN'05)*, 2005, pp. 114–121.

[4] Q. Niu, "Study and implementation of a improved group key protocol for mobile ad hoc networks," in *Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD 2007)*, vol. 1, July 2007, pp. 304–308.

[5] L. Luo, R. Safavi-Naini, J. Baek, and W. Susilo, "Self-organised group key management for ad hoc networks," in *ACM Symposium on Information, computer and communications security (ASIACCS'06)*, Mar. 2006, pp. 138–147.

[6] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE Symposium on Security and Privacy*, May 2003, pp. 197–213.

[7] E. Konstantinou, "Cluster-based group key agreement for wireless ad hoc networks," in *Third International Conference on Availability, Reliability and Security (ARES 08)*, Mar. 2008, pp. 550–557.

[8] T. Clausen and P. Jacquet, *Optimized Link State Routing Protocol (OLSR)*, RFC 3626, Oct. 2003.

[9] M. S. Bouassida, I. Chrisment, and O. Festor, "Efficient group key management protocol in MANETs using the multipoint relaying technique," in *Intl. Conference on Networking, Intl. Conference on Systems and Intl. Conference on Mobile Communications and Learning Technologies (ICN/ICONS/MCL 2006)*, Apr. 2006, pp. 64 – 71.

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

[11] 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.

[12] C. A. Maziero, "ARP: Petri net analyzer," Control and Microinformatic Lab., Federal Univ. Santa Catarina, Santa Catarina, Brazil, 1990.

[13] D. Lamch, "Verification and analysis of properties of dynamic systems based on petri nets," in *International Conference on Parallel Computing in Electrical Engineering (PARELEC'02)*, 2002, pp. 92–94.

[14] D. W. Carman, P. S. Kruus, and B. J. Matt, "Constraints and approaches for distributed sensor network security (final)," NAI Labs, Tech Report 00-010, Sept. 2000.

[15] M. E. M. Campista, I. M. Moraes, P. Esposito, A. Amodei Jr., L. H. M. K. Costa, and O. C. M. B. Duarte, "The ad hoc return channel: a low-cost solution for brazilian interactive digital TV," *IEEE Communications Magazine*, vol. 45, no. 1, pp. 136–143, Jan. 2007.