Modelling Adversaries and Security Objectives for Routing Protocols in Wireless Sensor Networks*

Gergely Ács, Levente Buttyán, and István Vajda
Laboratory of Cryptography and Systems Security (CrySyS)
Department of Telecommunications
Budapest University of Technology and Economics, Hungary
{acs, buttyan, vajda}@crysys.hu

Abstract

The literature is very broad considering routing protocols in wireless sensor networks (WSNs). However, security of these routing protocols has fallen beyond the scope so far. Routing is a fundamental functionality in wireless networks, thus hostile interventions aiming to disrupt and degrade the routing service have a serious impact on the overall operation of the entire network. In order to analyze the security of routing protocols in a precise and rigorous way, we propose a formal framework encompassing the definition of an adversary model as well as the "general" definition of secure routing in sensor networks. Both definitions take into account the feasible goals and capabilities of an adversary in sensor environments and the variety of sensor routing protocols. In spirit, our formal model is based on the simulation paradigm that is a successfully used technique to prove the security of various cryptographic protocols. However, we also highlight some differences between our model and other models that have been proposed for wired networks. Finally, we illustrate the practical usage of our model by presenting the formal description of a simple attack against an authenticated routing protocol, which is based on the well-known TinyOS routing.

1 Introduction

Routing is a fundamental function in every network that is based on multi-hop communications, and wireless sensor networks are no exceptions. Consequently, a multitude of routing protocols have been proposed for sensor networks in the recent past. However, most of these protocols have not been designed with security requirements in mind. This means that they can badly fail in hostile environments. Paradoxically, research on wireless sensor networks have been mainly fuelled by their potential applications in military settings where the environment is hostile. The natural question that may arise is why then security of routing protocols for sensor networks has fallen beyond the scope of research so far.

We believe that one important reason for this situation is that the design principles of secure routing protocols for wireless sensor networks are poorly understood today. First of all, there is no clear definition of what secure routing should mean in this context. Instead, the usual approach, exemplified in [10], is to list different types of possible attacks against routing in wireless sensor networks, and to define routing security implicitly as resistance to (some of) these attacks. However, there are several problems with this approach. For instance, a given protocol may resist a different set of attacks than another one. How to compare these protocols? Shall we call them both secure routing protocols? Or on what grounds should we declare one protocol more secure

than another? Another problem is that it is quite difficult to carry out a rigorous analysis when only a list of potential attack *types* are given. How can we be sure that all possible attacks of a given type has been considered in the analysis? It is not surprising that when having such a vague idea about what to achieve, one cannot develop the necessary design principles. It is possible to come up instead with some countermeasures, similar to the ones described in [10], which are potentially usefully to thwart some specific types of attacks, but it remains unclear how to put these ingredients together in order to obtain a secure and efficient routing protocol at the end.

In order to remedy this situation, we propose to base the design of secure routing protocols for wireless sensor networks on a formal security model. While the benefit of formal models is not always clear (indeed, in some cases, they tend to be overly complicated compared to what they achieve), we have already demonstrated their advantages in the context of ad hoc network routing protocols. More specifically, we developed formal security models in [4, 1, 2], and we successfully used them to prove the security of some ad hoc network routing protocols, and to find security holes in others. The idea here is to use the same approach in the context of wireless sensor networks. The rationale is that routing protocols in sensor networks are somewhat similar to those in ad hoc networks, hence they have similar pitfalls and they can be modeled in a similar way.

Thus, in this paper, we present a formal model, in which security of routing is precisely defined, and which can serve as the basis for rigorous security analysis of routing protocols proposed for wireless sensor networks. Our model is based on the simulation paradigm, where security is defined in terms of indistinguishability between an ideal-world model of the system (where certain attacks are not possible by definition) and the real-world model of the system (where the adversary is not constrained, except that he must run in time polynomial). This is a standard approach for defining security, however, it must be adopted carefully to the specific environment of wireless sensor networks.

Similar to [4], in this paper, we develop an adversary model that is different from the standard Dolev-Yao model, where the adversary can control all communications in the system. In wireless sensor networks, the adversary uses wireless devices to attack the systems, and it is more reasonable to assume that the adversary can interfere with communications only within its power range. In addition, we must also model the broadcast nature of radio communications.

However, in addition to the model described in [4], here we take into account that there are some attacks which exploit the constraint energy supply of sensor nodes (e.g., the adversary decreases the network lifetime by diverting the traffic in order to overload, and thus, deplete some sensor nodes). Hence, we explicitly model the energy consumption caused by sending a message between each pair of nodes in the network.

Another difference with respect to the model of [4] lies in the definition of the outputs of the ideal-world and the real-world models. It is tempting to consider the state stored in the routing tables of the nodes as the output, but an adversary can distort that state in unavoidable ways. This means that if we based our definition of security on the indistinguishability of the routing states in the ideal-world and in the real-world models, then no routing protocol would satisfy it. Hence, we define the output of the models as a suitable function of the routing state, which hides the unavoidable distortions in the states. This function may be different for different types of routing protocols, but the general approach of comparing the outputs of this function in the ideal-world and in the real-world models remain the same. For instance, this function could be the average length of the shortest pathes between the sensor nodes and the base station; then, even if the routing tables of the nodes would not always be the same in the ideal-world and in the real-world models, the protocol would still be secure given that the difference between the distributions of the average length of the shortest pathes in the two models is negligibly small.

The rest of the paper is organized as follows: In Section 2, we present the elements of our formal model, which includes the presentation of the adversary model adopted to wireless sensor networks, the description of the ideal-world and the real-world models, the general definition of the output of these models, as well as the definition of routing security. Then, in Section 3, we illustrate the usage of our model by representing in it a known insecurity of an authenticated version of the TinyOS routing protocol. Finally, in Section 4, we report on some related work,

and in Section 5, we conclude the paper.

We must note that the work described in this paper is a *work in progress*, and it should be considered as such. In particular, the reader will not find security proofs in this paper. There are two reasons for this: first, we are still developing the proof techniques, and second, we have not identified yet any routing protocols that would be secure in our model.

2 The model of wireless sensor networks

2.1 Adversary model

The adversary is represented by adversarial nodes in the network. An adversarial node can correspond to an ordinary sensor node, or a more resourced laptop-class device. In the former case, the adversary may deploy some corrupted sensor-class devices or may capture some honest sensor nodes. In the latter case, he has a laptop-class device with a powerful antenna and unconstrained energy supply. All of these adversarial nodes may be able to communicate in out-of-band channels (e.g., other frequency channel or direct wired connection), which may be used to create wormholes.

In general, when capturing honest sensor nodes, the adversary may be able to compromise their cryptographic secrets (assuming that such secrets are used in the system). However, in this paper, we assume that the adversary *cannot* compromise cryptographic material. This is certainly a simplifying assumption, and we intend to relax it in our future work.

The adversary attacking the routing protocol primarily intends to shorten the network lifetime, degrade the packet delivery ratio, increase his control over traffic, and increase network delay. Some of these goals are highly correlated; e.g., increasing hostile control over traffic may also cause the network delay to be increased.

In order to achieve the aforementioned goals, the adversary is able to perform simple message manipulations: fabricated message injection, message deletion, message modification and re-ordering of message sequences. In the followings, we describe how the adversary can perform message deletion and injection in a wireless sensor network. Re-ordering of message sequences is straightforward using message deletion and insertion, thus, we do not elaborate it further.

Basically, an adversarial node can affect the communication of two honest nodes in two cases: In the first case, an adversarial node relays messages between honest nodes which are not able to communicate directly with each other. In the second case, the honest nodes can also reach each other, and the adversarial node can also hear the nodes' communication, i.e., he can send and receive messages to/from both honest nodes. We further assume that communication range implies interference range, and vice-versa.

In case of adversarial relaying of messages between the nodes, all of the message manipulations are quite straightforward. On the contrary, if the honest nodes can also communicate with each other, message manipulations must be performed in a very sophisticated way. The adversarial node can inject messages easily, but deletion and modification require jamming capability. Message deletion may be achieved by employing various selective jamming techniques against either the sender node or the receiver node. Message modification is only feasible, if both the sender and the receiver nodes are within the communication range of the adversarial node. Here, we sketch two scenarios for message modification, which are illustrated on Figure 1. By these simple examples, we intend to point out the feasibility of message modification assuming even direct communication between the sender and the receiver node.

Scenario 1: There are two honest nodes X and Y, and node X intends to send a message m to node Y. A_1 and A_2 are adversarial nodes, where A_2 is able to interfere with Y's communication, but not with X's and A_1's communication. Let A_1 be in the communication range of X and Y, whereas A_2 can only communicate with Y. When X transmits m to Y, node A_1 overhears m, meanwhile A_2 performs jamming to cause Y not to be able to receive m. In order to take this action, A_1 and A_2 are connected by an out-of-band channel, thus, A_1 can send a signal to A_2 when A_2 should start jamming Y's communication. It is also feasible that A_2 performs constant jamming for a certain amount of time, afterwards, A_1 can send the modified message m' to Y.

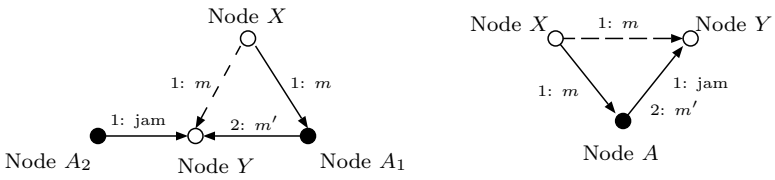

Figure 1: Message modification performed by the cooperation of two adversarial nodes A_1 and A_2 (on the right-hand side) in Scenario 1, and employing overhearing, jamming, and relaying with a single adversarial node A (on the left-hand side) in Scenario 2. Honest nodes are labelled by X and Y. Arrows between nodes illustrate the direction of communication, the sequence of message exchanges are also depicted on these arrows. Dashed arrows illustrate failed message delivery caused by jamming.

Scenario 2: In this scenario, there is only one adversarial node denoted by A. We assume that transmitting a message from the routing sublayer consists of passing the message to the data-link layer, which, after processing the message, also passes it further to the physical layer. The data-link layer uses CRC in order to provide some protection against faults in noisy channels; a sender generally appends a frame check sequence to each frame (e.g., see [7]). The adversary can exploit this CRC mechanism to modify a message in the following way (illustrated on Figure 1). When X transmits message m to Y, node A also overhears m, in particular, he can see the frame(s) belonging to m. A intends to modify message m. Here, we must note that most messages originated from the routing sublayer are composed of only one frame per message in the data-link layer due to performance reasons, especially when they are used to discover routing topology. Upon reception of the frame corresponding to the message, the adversary can corrupt the frame check sequence by jamming once the data field of the frame has been received. This causes node Y to drop the frame (and the message), since Y detects that the last frame is incorrect, and waits for retransmission. At this point, if some acknowledgement mechanism is in use, A should send an acknowledgement to X so that it does not re-send the original frame. In addition, A retransmits message m' in the name of X, where m' is the modified message.

The feasibility of jamming attacks is studied and demonstrated in [17]. Although, the authors conclude in that paper that the success of jamming attacks mainly depend on the distance of the honest nodes and the jammer node, various jamming techniques has been presented there that can severely interfere with the normal operation of the network.

2.2 Network model

We assume that each honest device has exactly one antenna in the network. If the adversary uses several antennas we represent each of them by a distinct node. The network nodes are considered to be static, and we further assume that there is a single base station in the network.

Let us denote the honest nodes in the network by $v_0, v_1, \ldots, v_k$, where v_0 denotes the base station. Similarly, $v_{k+1}, \ldots, v_{k+m}$ represent the adversarial nodes. The set of all nodes is denoted by V. Furthermore, n denotes the number of all nodes in the network, i.e., $n = |V| = k + m + 1$. For each pair of nodes v_i and v_j, we define e_{v_i,v_j} to be the energy level needed to transmit a message from v_i to v_j, where $v_i, v_j \in V$. This values can be ordered in a matrix with size $n \times n$, called *reachability matrix*, and it is denoted by $\underline{\underline{E}}$.[1] In the rest, if we intend to emphasize the distinction between the honest and the adversarial nodes in the notation, we prefer to denote the adversarial nodes by $v_1^*, \ldots, v_m^*$ (where $v_\ell^* = v_{k+\ell}$, $1 \leq \ell \leq m$).

For the sake of simplicity, we also assume that at least energy e_{v_i,v_j} is needed for node v_i to interfere with node v_j's packet reception. This means that if v_i can reach v_j, then v_i can also interfere with all the communication of v_j.

Let us assume that each node uses a globally unique identifier in the network, and these identifiers are authenticated in some way (e.g., by symmetric keys). We denote the set of these

[1] In this paper, the rows and the columns of all matrices are numbered from zero.

identifiers by L, and there is a function $\mathcal{L} : V \to L \cup \{\mathsf{undef}\}$ that assigns an identifier to each node, where $\mathsf{undef} \notin L$. According to our adversary model described in Subsection 2.1, we assume that the adversary has no (authenticated) identifier in the network, i.e., $\mathcal{L}(v_j^*) = \mathsf{undef}$ for all $1 \leq j \leq m$.

We also introduce a cost function $\mathcal{C} : V \to \mathbb{R}$, which assigns a cost to each node (e.g., the remaining energy in the battery, or constant 1 to each node in order to represent hop-count).

Configuration: A configuration *conf* is a quadruple $(V, \mathcal{L}, \underline{E}, \mathcal{C})$ that consists of the the set of nodes, the labelling function, the reachability matrix, and the cost function of nodes.

2.3 Security objective function

Diverse sensor applications entail different requirements for routing protocols. For instance, remote surveillance applications may require minimal delay for messages, while sensor applications performing some statistical measurements favour routing protocols prolonging network lifetime. The diversity of routing protocols is caused by these conflicting requirements: e.g., shortest-path routing algorithms cannot maximize the network lifetime, since always choosing the same nodes to forward messages causes these nodes to run out of their energy supply sooner. Several sensor routing protocols use a trade-off to satisfy conflicting requirements [16, 11].

This small argument also points out that one cannot judge the utility of all routing protocols uniformly. Without a unified metric of utility we cannot refine our security objectives for routing protocols. By the above argument, a routing protocol that is secure against attacks aiming at decreasing network-lifetime cannot be secure against attacks aiming at increasing network delay. We model the negatively correlated requirements of routing, and essentially, our security objectives in a very general manner. We represent the output of a routing protocol, which is actually the ensemble of the routing entries of the honest nodes, with a given configuration *conf* by a matrix $\underline{T}^{conf}$ with size $k+1 \times k+1$.[2] $\underline{T}^{conf}_{i,j} = 1$, if honest node v_i sends every message to an honest node identified by $\mathcal{L}(v_j)$ in order to deliver the message to the base station, otherwise let $\underline{T}^{conf}_{i,j}$ be 0. In the rest of the paper, we shortly refer to the result of a routing protocol with a given configuration as a *routing topology*, which can be considered as a directed graph described by matrix $\underline{T}^{conf}$. In the following, we will omit the index *conf* of $\underline{T}$ when the configuration can be unambiguously determined in a given context. In fact, $\underline{T}^{conf}$ is a random variable, where the randomness is caused by the sensor readings initiated randomly by the environment, processing and transmission time of the sensed data, etc.

Let us denote the set of all configurations by $\mathbb{G}$. Furthermore, $\mathbb{T}$ denotes the set of the routing topologies of all configurations. The security objective function $\mathcal{F} : \mathbb{G} \times \mathbb{T} \to \mathbb{R}$ assigns a real number to a random routing topology of a configuration. This function intends to distinguish "attacked" topologies from "non-attacked" topologies based on a well-defined security objective. We note that the definition of $\mathcal{F}$ is protocol dependent. For example, let us consider routing protocols that build a routing tree, where the root is the base station. We can compare routing trees based on network lifetime by the following security objective function

$$\mathcal{F}(conf, \underline{T}^{conf}) = \frac{1}{k} \sum_{i=1}^{k} \mathcal{E}(v_i, conf, \underline{T}^{conf})$$

where $\mathcal{E} : V \times \mathbb{G} \times \mathbb{T} \to \mathbb{R}$ assigns the overall energy consumption of the path from a node v_i to v_0 (the base station) in a routing tree of a configuration. Since $\underline{T}^{conf}$ is a random variable, the output of $\mathcal{F}$ is a random variable too. If the distribution of this output in the presence of an attacker non-negligibly differs from the distribution when there's no attacker, then the protocol is not secure. If we intend to compare routing trees based on network delay a simple security

[2] Of course, here we only consider the result of the protocol with respect to the honest nodes, since the adversarial nodes may not follow the protocol rules faithfully.

objective function may be

$$\mathcal{F}(conf, \underline{T}^{conf}) = \frac{1}{k} \sum_{i=1}^{k} \mathcal{M}(v_i, conf, \underline{T}^{conf})$$

where $\mathcal{M} : V \times \mathbb{G} \times \mathbb{T} \to \mathbb{R}$ assigns the length of the path from a node to v_0 in a routing topology of a configuration.

2.4 Dynamic model

Following the simulation paradigm, we define a real-world model and an ideal-world model. The real-world model represents the real operation of the protocol and the ideal-world model describes how the system should work ideally. Both models contain an adversary. The real-world adversary is not constrained apart from requiring it to run in time polynomial. This enables us to be concerned with arbitrary feasible attacks. In addition, the ideal-world adversary is constrained in a way that it cannot modify messages and inject extra ones due to the construction of the ideal-world system. In other words, all attacks that modify or inject any messages is unsuccessful in the ideal-world system. However, the ideal-world adversary can perform attacks that are unavoidable or very costly to defend against (e.g., message deletion).

Once the models are defined, the goal is to prove that for any real-world adversary, there exist an ideal-world adversary that can achieve essentially the same effects in the ideal-world model as those achieved by the real-world adversary in the real-world model (i.e., the ideal-world adversary can simulate the real-world adversary).

2.4.1 Real-world model

The real-world model that corresponds to a configuration $conf = (V, \mathcal{L}, \underline{E}, \mathcal{C})$ and adversary $\mathcal{A}$ is denoted by $sys^{\mathsf{real}}_{conf,\mathcal{A}}$, and it is illustrated on Figure 2. We model the operation of the protocol participants by interactive and probabilistic Turing machines. Correspondingly, we represent the adversary, the honest sensor nodes, and the broadcast nature of the radio communication by machines A, M_i, and C, respectively. These machines communicate with each other via common tapes.

Each machine must be initialized with some input data (e.g., cryptographic keys, reachability matrix, etc.), which determines its initial state. Moreover, the machines are also provided with some random input (the coin flips to be used during the operation). Once the machines have been initialized, the computation begins. The machines operate in a reactive manner, i.e., they need to be activated in order to perform some computation. When a machine is activated, it reads the content of its input tapes, processes the received data, updates its internal state, writes some output on its output tapes, and goes back to sleep. The machines are activated in rounds by a hypothetic scheduler, and each machine in each round is activated only once. The order of activation is arbitrary with the only restriction that C must be activated at the end of the rounds.

Now, we present the machines in more details:

- Machine C. This machine is intended to model the radio communication. It has input tapes out_i and out_j^*, from which it reads messages written by M_i and A, resp. It also has output tapes in_i and in_j^*, on which it writes messages to M_i and A, resp. C is also initialized by matrix $\underline{E}$ at the beginning of the computation.

  Messages on tape out_i can have the format $(\ell_{sndr}, cont, e, dest)$, where $\ell_{sndr} \in L$ is the identifier of the sender, $cont$ is the message content, e is the energy level to be used to determine the range of transmission, and $dest$ is the identifier of the intended destination $dest \in L \cup \{*\}$, where $*$ indicates broadcast message.

  Messages on tape out_j^* can have the following formats:

– $(\mathsf{MSG}, \ell_{sndr}, cont, e, dest)$: MSG message models a normal broadcast message sent by the adversary to machine C with sender identifer $\ell_{sndr} \in L$, message content $cont$, energy level e, and identifier of the intended destination $dest \in L \cup \{*\}$.

– (JAM, e): Special JAM message, that is sent by the adversary to machine C, models the jamming capability of the adversary. When machine C receives a message JAM, it performs the requested jamming by deleting all messages in the indicated range e around the jamming node, which means that those deleted messages are not delivered to the nodes (including the jammer node itself) within the jamming range.

– $(\mathsf{DEL}, \ell_{tar}, e)$: Special DEL message, that is sent by the adversary to machine C, models the modification capability of the adversary. When receiving a message DEL with identifier $\ell_{tar} \in L$, machine C does not deliver any messages sent by node $v' \in V$, where $\mathcal{L}(v') = \ell_{tar}$, if v' is within the indicated range e, except the adversarial node itself that will receive the deleted messages. This models the sophisticated jamming technique that we described in Subsection 2.1.

In a more formal way, when reading a message $msg^*_{in} = (\mathsf{MSG}, \ell_{sndr}, cont, e, dest)$ from out^*_j, C determines the nodes which receive the message by calculating the set of nodes $V_e \subseteq V$, such that for all $v' \in V_e$ $e_{v_j,v'} \leq e$. Finally, C processes msg^*_{in} as follows.

1. if $dest \in L \cup \{*\}$, then C writes
   – $msg_{out} = (\ell_{sndr}, cont, dest)$ to the input tapes of machines corresponding to honest nodes in V_e
   – $msg^*_{out} = (\mathsf{MSG}, \ell_{sndr}, cont, dest)$ to the input tapes of machines corresponding to adversarial nodes in $V_e \setminus \{v^*_j\}$
2. otherwise C discards msg^*_{in}

When reading a message $msg^*_{in} = (\mathsf{JAM}, e)$ from out^*_j, C determines the set of nodes which receive the message by calculating $V_e \subseteq V$, such that for all $v' \in V_e$ $e_{v_j,v'} \leq e$. Afterwards, C does not write any messages within the same round to the input tapes of machines corresponding to V_e.

When reading a message $msg^*_{in} = (\mathsf{DEL}, \ell_{tar}, e)$ from out^*_j, C determines the set of nodes which receive the message by calculating $V_e \subseteq V$, such that for all $v' \in V_e$ $e_{v_j,v'} \leq e$. Finally, C processes msg^*_{in} as follows.

1. if there exists $v_x \in V_e$ $(1 \leq x \leq k)$, such that $\mathcal{L}(v_x) = \ell_{tar}$, then C does not write any messages within the same round from tape out_x to the input tapes of machines corresponding to $V_e \setminus \{v^*_j\}$
2. otherwise C discards msg^*_{in}

When reading a message $msg_{in} = (\ell_{sndr}, cont, e, dest)$ from out_i, C determines the set of nodes which receive the message by calculating $V_e \subseteq V$, such that for all $v' \in V_e$ $e_{v_j,v'} \leq e$. Finally, C processes msg_{in} as follows.

1. if $dest \in L \cup \{*\}$, then C writes
   – $msg_{out} = (\ell_{sndr}, cont, dest)$ to the input tapes of machines corresponding to honest nodes in $V_e \setminus \{v_i\}$
   – $msg^*_{out} = (\mathsf{MSG}, \ell_{sndr}, cont, dest)$ to the input tapes of machines corresponding to adversarial nodes in V_e
2. otherwise C discards msg_{in}

- Machine M_i. This machine models the operation of honest sensor nodes, and it corresponds to node v_i. It has input tape in_i and output tape out_i, which are shared with machine C. The format of input messages must be $(\ell_{sndr}, cont, dest)$, where $dest \in L \cup \{*\}$. The format

of output messages must be $(\ell_{sndr}, cont, e, dest)$, where ℓ_{sndr} must be $\mathcal{L}(v_i)$, $dest \in L \cup \{*\}$, and e indicates the transmission range of the message for C. When this machine reaches one of its final states or there is a time-out during the computation process, it outputs its routing table.

- Machine A. This machine models the adversary logic. Encapsulating each adversarial node into a single machine allows us to model wormholes inside A. One can imagine that the adversary deploy several antennas in the network field, which are connected to a central adversary logic. In this convention, node v_j^* corresponds to an adversarial antenna, which is modelled by input tape in_j^* and output tape out_j^*. These tapes are shared with machine C.

  The format of input messages must be $msg_{in}^* = (\mathsf{MSG}, \ell_{sndr}, cont, e, dest)$, where $dest \in L \cup \{*\}$.

  The format of output messages msg_{out}^* can be

  - $(\mathsf{MSG}, \ell_{sndr}, cont, e, dest)$, where $dest \in L \cup \{*\}$ and e indicates the transmission range of the message;
  - (JAM, e), where e indicates the range of jamming;
  - $(\mathsf{DEL}, \ell_{tar}, e)$, where e indicates the range of selective jamming, and $\ell_{tar} \in L$.

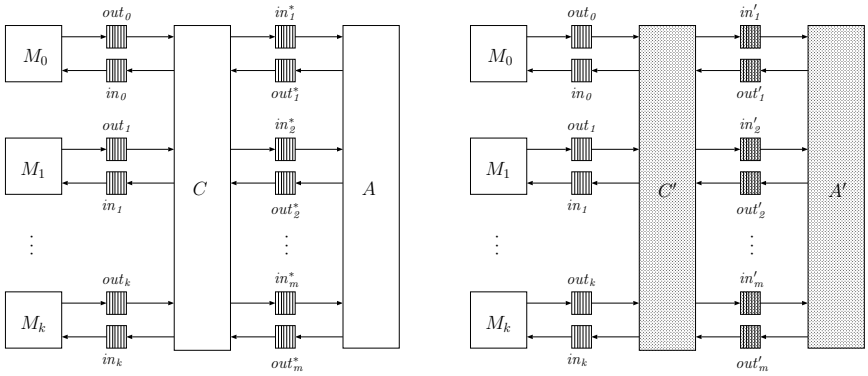

Figure 2: The real-world model (on the left-hand side) and the ideal-world model (on the right-hand side).

The computation ends, when all machines M_i reach their final states, or there is a time-out. The output of $sys_{conf,\mathcal{A}}^{\mathsf{real}}$ is the value of the security objective function $\mathcal{F}$ applied to the resulted routing topology defined in Subsection 2.3 and configuration $conf$. The routing topology is represented by the ensemble of the routing entries of machines M_i. We denote the output by $Out_{conf,\mathcal{A}}^{\mathsf{real},\mathcal{F}}(r)$, where r is the random input of the model. In addition, $Out_{conf,\mathcal{A}}^{\mathsf{real},\mathcal{F}}$ will denote the random variable describing $Out_{conf,\mathcal{A}}^{\mathsf{real},\mathcal{F}}(r)$ when r is chosen uniformly at random.

2.4.2 Ideal-world model

The ideal-world model (illustrated on Figure 2) that corresponds to a configuration $conf = (V, \mathcal{L}, \underline{E}, \mathcal{C})$ and adversary $\mathcal{A}'$ is denoted by $sys_{conf,\mathcal{A}'}^{\mathsf{ideal}}$. The ideal-world model is identical to the real-world model with the exception that the ideal-world adversary cannot modify and inject extra messages. However, he is allowed to simply drop any messages or perform jamming, since these attacks are unavoidable, or at least, they are too costly to defend against. Our model is considered to be ideal in this sense. Comparing to the real-world model, we replace machine C with machine C' and machine A with machine A' in order to implement our restricted ideal-world adversary. Hence, we only detail the operation of C' and A' here, since M_i are the same as in the real-world model.

Receiving an MSG message from machines M_i, C' internally stores that message with a unique message identifier in its internal store. Delivering any MSG message to A', C' also includes the message identifier into the message. A' can send an MSG message to C' with a different format; it only contains an identifier id and an energy level e. Upon the reception of such a message, C' searches for the original message which is associated with identifier id in its internal store, and delivers this stored message using the energy level e. Although A' also receives the original message with its associated identifier from C', he is not able to modify that, since C' only accepts a message identifier issued by himself and an energy level from A'. In other words, A' can only delete messages, since A' can also send special DEL and JAM messages to C'. We elaborate the operation of C' and A' in a more formal way as follows.

A' and C' communicate via tapes in'_j and out'_j.

- Machine C'. It has input tapes out_i and out'_j, from which it reads messages written by M_i and A, resp. It also has output tapes in_i and in'_j, on which it writes messages to M_i and A, resp. C' is also initialized by matrix $\underline{E}$. In addition, it sets its internal variable $id_{C'}$ to 1 at the beginning of the computation.

  C' interacts with machines M_i in a similar way as C does in the real-world model; when reading a message $msg_{in} = (\ell_{sndr}, cont, e, dest)$ from out_i, C' processes msg_{in} identically to C in the real-world model only with one exception: Before writing $msg'_{in} = (\mathsf{MSG}, id_{C'}, \ell_{sndr}, cont, dest)$ to output tapes in'_j, C' internally stores msg'_{in} in set S. After writing msg'_{in} to output tapes in'_j, C' increments $id_{C'}$ by one. Therefore, C' knows what messages are passed to A from M_i. Messages on out'_j can have the formats:

  - (MSG, id, e): MSG message models a normal broadcast message sent by the ideal-world adversary to machine C', where e indicates the transmission range of the message identified by id.
  - (JAM, e): Special JAM message, that is sent by the adversary to machine C, models the jamming capability of the ideal-world adversary, where e indicates the range of jamming.
  - $(\mathsf{DEL}, \ell_{tar}, e)$: Special DEL message, that is sent by the adversary to machine C, models the modification capability of the ideal-world adversary, where e indicates the range of selective jamming, and $\ell_{tar} \in L$.

  When reading a message $msg'_{in} = (\mathsf{MSG}, id, e)$ from out'_j, machine C' operates differently from C. C' determines the set of nodes which receive the message by calculating $V_e \subseteq V$, such that for all $v' \in V_e$ $e_{v_j,v'} \leq e$. Finally, C' processes msg'_{in} as follows.

  1. if $1 \leq id \leq id_{C'}$, then C' searches the $msg' = (\mathsf{MSG}, id', \ell'_{sndr}, cont', dest')$ in S such that id' equals to id, and C writes
     - $msg_{out} = (\ell'_{sndr}, cont', dest')$ to the input tapes of machines corresponding to honest nodes in V_e
     - $msg'_{out} = (\mathsf{MSG}, id', \ell'_{sndr}, cont', dest')$ to the input tapes of machines corresponding to adversarial nodes in $V_e \setminus \{v_j^*\}$
  2. otherwise C' discards msg^*_{in}

  When reading a message $msg'_{in} = (\mathsf{JAM}, e)$ or $msg'_{in} = (\mathsf{DEL}, \ell_{tar}, e)$ from out'_j, machine C' operates the same way as C does in case of the corresponding message formats.

- Machine A'. It has output tapes out'_j and input tapes in'_j. The format of messages on input tape in'_j must be $msg'_{in} = (\mathsf{MSG}, id, \ell_{sndr}, cont, e, dest)$, where $dest \in L \cup \{*\}$.

  The format of output messages msg'_{out} can be

  - (MSG, id, e), where id is a message identifier and e indicates the transmission range of the message identified by id;

– (JAM, e), where e indicates the range of jamming;
– $(\mathsf{DEL}, \ell_{tar}, e)$, where e indicates the range of selective jamming, and $\ell_{tar} \in L$.

The computation ends, when all machines M_i reach their final states, or there is a time-out. Similar to the real-world model, the output of $sys^{\mathsf{ideal}}_{conf,\mathcal{A}}$ is the value of the security objective function $\mathcal{F}$ applied to the resulted routing topology and configuration $conf$. The routing topology is represented by the ensemble of the routing entries of machines M_i. We denote the output by $Out^{\mathsf{ideal},\mathcal{F}}_{conf,\mathcal{A}'}(r)$, where r is the random input of the model. Moreover, $Out^{\mathsf{ideal},\mathcal{F}}_{conf,\mathcal{A}'}$ will denote the random variable describing $Out^{\mathsf{ideal},\mathcal{F}}_{conf,\mathcal{A}'}(r)$ when r is chosen uniformly at random.

2.5 Definition of secure routing

Let us denote the security parameter of the model by κ (e.g., κ is the key length of the cryptographic primitive employed in the routing protocol, such as digital signature, MAC, etc.). Based on the model described in the previous subsections, we define routing security as follows:

Definition 1 (Statistical security) *A routing protocol is statistically secure with security objective function $\mathcal{F}$, if for any configuration conf and any real-world adversary $\mathcal{A}$, there exists an ideal-world adversary $\mathcal{A}'$, such that $Out^{\mathsf{real},\mathcal{F}}_{conf,\mathcal{A}}$ is statistically indistinguishable from $Out^{\mathsf{ideal},\mathcal{F}}_{conf,\mathcal{A}'}$. Two random variables are statistically indistinguishable if the L_1 distance of their distributions is a negligible function of the security parameter κ.*

Intuitively, if a routing protocol is statistically secure, then any system using this routing protocol cannot satisfy its security objectives represented by function $\mathcal{F}$ only with a probability that is a negligible function of κ.

This negligible probability is related to the fact that the adversary can always forge the cryptographic primitives (e.g., generate a valid digital signature) with a very small probability depending on the value of κ.

3 Insecurity of TinyOS routing

In this section, we present an authenticated routing mechanism based on the well-known TinyOS routing, and we show that it is not secure in our model for a given security objective function representing a very minimal security requirement.

3.1 Operation of an authenticated routing protocol

Originally, the authors of TinyOS implemented a very simple routing protocol, where each node uses a globally unique identifier. The base station periodically initiates a routing topology discovery by flooding the network by a beacon message. Upon reception of the first beacon within a single beaconing interval, each sensor node stores the identifier of the node, from which it received the beacon, as its parent (aka. next-hop towards the base station), and then re-broadcasts the beacon after changing the sender identifier to its own identifier. As for each node only one parent is stored, the resulted routing topology is a tree. Every sensor node receiving a data packet forwards that towards the base station by sending the packet to its parent. A lightweight cryptographic extension is employed in [14] in order to authenticate the beacon by the base station. This authenticated variant of TinyOS routing uses μTesla scheme to provide integrity for the beacon; each key is disclosed by the next beacon in the subsequent beaconing interval. We remark that this protocol has only been defined informally that inspired us to present a new protocol, which provides the "same" security as the authenticated routing protocol in [14], but due to its simplicity it fits more in demonstrating the usage of our model. Consequently, the presented attack against this new protocol also works against the protocol in [14]. We must note again that this protocol is only intended to present the usefulness of our model rather than to be considered as a proposal of a new sensor routing protocol.

We assume that the base station B has a public-private key pair, where the public key is denoted by K_{pub}. Furthermore, it is assumed that each sensor node is also deployed with K_{pub}, and they are capable to perform digital signature verification with K_{pub} as well as to store some beacons in its internal memory. We note that B never relays messages between sensor nodes.

Initially, B creates a beacon, that contains a constant message identifier BEACON, a randomly generated number rnd, the identifer of the base station Id_B, and a digital signature sig_B generated on the previous elements except Id_B. Afterwards, the base station floods the network by broadcasting this beacon:

$$B \rightarrow * \quad : \quad msg_1 = (\mathsf{BEACON}, \mathsf{rnd}, Id_B, \mathsf{sig}_B)$$

Each sensor node X receiving msg_1 checks whether it has already received a beacon with the same rnd in conjuction with a correct signature before. If it is true, the node discards msg_1, otherwise it verifies sig_B. If the verification is successful, then X sets Id_B as its parent, stores msg_1 in its internal memory, and re-broadcasts the beacon by changing the sender identifier Id_B to its own identifier Id_X:

$$X \rightarrow * \quad : \quad msg_2 = (\mathsf{BEACON}, \mathsf{rnd}, Id_X, \mathsf{sig}_B)$$

If the signature verification is unsuccessful, then X discards msg_1. Every sensor node receiving msg_2 performs the same steps what X has done before.

Optionally, B can initiate this topology construction periodically by broadcasting a new beacon with different rnd.

In the rest, we shortly refer to this protocol as ABEM (Authenticated Beaconing Mechanism).

3.2 Formalization of a simple attack

A simple security objective is to guarantee the correctness of all routing entries in the network. Namely, it is desirable that a sender node v_i is always able to reach node v_j, if v_i set $\mathcal{L}(v_j)$ as its parent identifier earlier. It means that if node v_i sets node $\mathcal{L}(v_j)$ as its parent identifier, then $E_{i,j}$ should contain a finite value, or v_i as well as v_j should have an adversarial neighboring node $v^*_{\ell_1}$ and $v^*_{\ell_2}$, resp., such that $\underline{E}_{i,k+\ell_1}$ and $\underline{E}_{k+\ell_2,j}$ are finite values, where $1 \leq \ell_1, \ell_2 \leq m$ and $\ell_1 \neq \ell_2$ may hold.

In order to formalize this minimal security requirement, we introduce the following security objective function

$$\mathcal{F}^{\mathrm{ABEM}}(conf, \underline{T}) = \begin{cases} 1, & \text{if } \forall i,j : \underline{T}_{i,j} \cdot \underline{E}'_{i,j} \cdot \left(\prod_{\ell=1}^{m} \underline{E}'_{i,k+\ell} + \prod_{\ell=1}^{m} \underline{E}'_{k+\ell,j}\right) = 0 \\ 0, & \text{otherwise} \end{cases}$$

where we derive matrix $\underline{E}'$ with size $n \times n$ from $\underline{E}$, so that $\underline{E}'_{i,j} = 1$, if $\underline{E}_{i,j} = \infty$, otherwise $\underline{E}'_{i,j} = 0$. In other words, $\underline{E}'_{i,j} = 1$, if v_i cannot send a message directly to v_j, otherwise $\underline{E}'_{i,j} = 0$.

We will show that ABEM is not secure in our model for security objective function $\mathcal{F}^{\mathrm{ABEM}}$. In particular, we present a configuration $conf$ and an adversary $\mathcal{A}$, for which there doesn't exist any ideal-world adversary $\mathcal{A}'$, such that $Out^{\mathsf{real},\mathcal{F}^{\mathrm{ABEM}}}_{conf,\mathcal{A}}$ is statistically indistinguishable from $Out^{\mathsf{ideal},\mathcal{F}^{\mathrm{ABEM}}}_{conf,\mathcal{A}'}$. Equivalently, we show that for a real-world adversary $\mathcal{A}$, $\mathcal{F}^{\mathrm{ABEM}}(conf, \underline{T}) = 0$ with a probability that is a non-negligible function of κ in the real-world model, while $\mathcal{F}^{\mathrm{ABEM}}(conf, \underline{T}') = 0$ with probability zero for every ideal-world adversary $\mathcal{A}'$ in the ideal-world model, where $\underline{T}'$ describes the routing topology in the ideal-world model. Moreover, the success probability of the real-world adversary $\mathcal{A}$ described below is independent from κ.

The configuration $conf$ and the result of the attack is depicted on Figure 3. We assume that the base station broadcasts only a single beacon during the computational process, i.e., only a single beaconing interval is analyzed in our model. At the beginning, the base station B floods the network by a beacon

$$B \rightarrow * \quad : \quad msg'_1 = (\mathsf{BEACON}, \mathsf{rnd}, B, \mathsf{sig}_B)$$

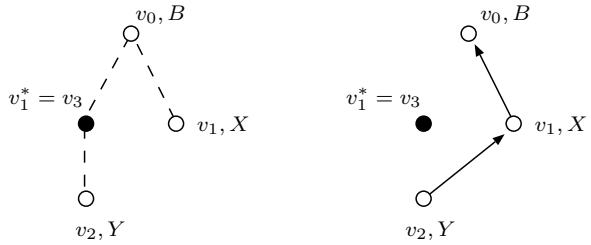

Figure 3: A simple attack against ABEM. v_0, v_1, and v_2 are honest nodes with identifiers $\mathcal{L}(v_0) = B$, $\mathcal{L}(v_1) = X$, and $\mathcal{L}(v_2) = Y$, whereas v_1^* is an adversarial node. $\underline{E}_{1,0}$, $\underline{E}_{3,0}$, $\underline{E}_{2,3}$ are finite values, and $\underline{E}_{3,1} = \underline{E}_{2,0} = \underline{E}_{2,1} = \infty$. Links are assumed to be symmetric, i.e., $\underline{E}_{i,j} = \underline{E}_{j,i}$. The configuration is illustrated on the left-hand side, where a dashed line denote a direct link. In the routing topology of the real-world model, on the right-hand side, v_2 sets X as its parent identifier, however, $\underline{E}_{2,1} = \infty$ and $\underline{E}_{3,1} = \infty$.

Both adversarial node v_1^* and honest node X receive this beacon, and X sets B as its parent, since the verification of the signature is successful. X modifies the beacon by replacing sender identifier B to X, and broadcasts the resulted beacon:

$$X \rightarrow * \quad : \quad msg_2' = (\mathsf{BEACON}, \mathsf{rnd}, X, \mathsf{sig}_B)$$

In parallel, v_1^* modifies the beacon by replacing sender identifier B to X, and broadcasts the resulted beacon:

$$v_1^* \rightarrow * \quad : \quad msg_2' = (\mathsf{BEACON}, \mathsf{rnd}, X, \mathsf{sig}_B)$$

Upon the reception of msg_2', node Y sets X as its parent, since sig_B is correct.

In the real-world model, these actions result $\underline{T}_{2,1} = 1$, which implies that $\mathcal{F}^{\mathrm{ABEM}}(conf, \underline{T}) = 0$. On the contrary, $\mathcal{F}^{\mathrm{ABEM}}(conf, \underline{T}')$ never equals to 0, where $\underline{T}'$ represents the routing topology in the ideal-world model. Let us assume that $\mathcal{F}^{\mathrm{ABEM}}(conf, \underline{T}') = 0$, which means that $\underline{T}'_{1,2} = 1$ or $\underline{T}'_{2,1} = 1$. $\underline{T}'_{1,2} = 1$ is only possible, if X receives

$$msg_3' = (\mathsf{BEACON}, \mathsf{rnd}, Y, \mathsf{sig}_B)$$

However, it yields contradiction, since $\underline{E}_{3,1} = E_{2,1} = \infty$, and B never broadcasts msg_3'. Similarly, if $\underline{T}'_{2,1} = 1$ then Y must receive msg_2', which means that v_1^* must broadcast msg_2'. Conversely, B never broadcasts msg_2', and $\underline{E}_{3,1} = \infty$. Therefore, v_1^* can only broadcast msg_2', if he successfully modifies msg_1' or forges msg_2'. However, it also contradicts our assumption that the ideal-world adversary cannot modify and inject messages in the ideal-world model.

4 Related work

In [10], the authors map some adversary capabilites and some feasible attacks against routing in wireless sensor networks, and they define routing security implicitly as resistance to (some of) these attacks. Hence, the security of sensor routing is only defined informally, and the counter-measures are only related to specific attacks. In this way, we even cannot compare the sensor routing protocols in terms of security. Another problem with this approach is the lack of a formal model, where the security of sensor routing can be described in a precise and rigorous way. While secure messaging and key-exchange protocols are classical and well-studied problems in traditional networks [3, 15], formal modelling of secure routing in sensor networks has not been considered so far. The adversarial nodes are also classified into the groups of sensor-class and laptop-class nodes in [10], but the capabilities of an adversarial node regarding message manipulations are not discussed.

The simulation paradigm is described in [15, 5]. These models were mainly proposed with wired networks in mind typically implemented on the well-known Internet architecture, and the wireless context is not focused there. In our opinion, the multi-hop nature of communications is an inherent characteristic of wireless sensor networks, therefore, it should be explicitly modelled. In more particular, the broadcast nature of communication enables a party to overhear the transmission of a message that was not destined to him, however, this transmission can be received only in a certain range of the sender. The size of this range is determined by the power at which the sender sent the message. Another deviation from [15] is the usage of the security objective function in the definition of security. In [15], the indistinguishability is defined on the view of the honest parties (on their input, states, and output) in the ideal-world and in the real-world models. However, an adversary can distort the states of the honest parties in unavoidable ways, and hence, the classical definition would be too strong and no routing protocol would satisfy it. On the other hand, our model is compliant with [15] considering high-level connections between nodes. In [15], the standard cryptographic system allows us to define each high-level connection as secure (private and authentic), authenticated (only authentic), and insecure (neither private nor authentic). In this taxonomy, the communication channel between two honest nodes can be either insecure or secure in our model. If an adversarial node is placed in the communication range of one of the communicating nodes, then it is considered to be an insecure channel. If the adversary can reach none of the communicating nodes, the channel between that nodes is hidden from the adversary, and thus, it is considered to be secure.

Although some prior works [18, 12] also used formal techniques to model the security of multi-hop routing protocols, these ones were mainly proposed for ad hoc routing. Moreover, the model proposed in [12] is based on CPAL-ES, and the model in [18] is similar to the strand spaces model. Both of these formal techniques differ from the simulation paradigm.

Our work is primarily based on [4, 1]. Here, the authors also use the simulation paradigm to prove the security of routing protocols in wireless ad-hoc networks. However, our model differs from the models in [4, 1] in two ways:

- *Adversary model*: The adversary in [4] and [1] is assumed to have the same resources and communcation capabilities as an ordinary node in the network. Therefore, that adversary model deviates from the so-called Dolev-Yao model in [6]. In our work, the adversary also uses wireless devices to attack the systems, and it is reasonable to assume that the adversary can interfere with communications only within its power range. The adversarial nodes belonging to the sensor-class nodes has the same resources and communication capabilities as an ordinary sensor node, but a more resourced adversarial node (e.g., laptops) may affect the overall communication of an entire part of the network depending on the power range of the resourced adversarial device. That resourced devices also make the adversary able to perform more sophisticated message manipulations.

- *Modelling security objectives*: In ad hoc networks, nodes contruct routes between a source and a destination [13, 8], whereas sensor nodes should build a complete routing topology for the entire network. In case of sensor networks, the only destination for all nodes is the base station [9]. In addition, sensor nodes are resource constrained, which implies that we also need to model the energy consumption of sensor nodes, since several attacks impacts the network lifetime. These differences from ad hoc networks has yielded a wide range of sensor applications, and thus, sensor routing protocols [9] are much diverse than ad hoc routing protocols. Hence, the security objectives cannot be modelled uniformly for sensor routing protocols.

5 Conclusion

In this paper, we proposed a formal security model for routing protocols in wireless sensor networks. Our model is based on the well-known simulation paradigm, but it differs from previously proposed models in several important aspects. First of all, the adversary model is carefully adopted to the

specific characteristics of wireless sensor networks. In our model, the adversary is not all-powerful, but it can only interfere with communications within its own radio range. A second important contribution is that we defined the output of the dynamic models that represent the ideal and the real operations of the system as a suitable function of the routing state of the honest nodes, instead of just using the routing state itself as the output. We expect that this will allow us to model different types of routing protocols in a common framework. In addition, this approach hides the unavoidable distortions caused by the adversary in the routing state, and in this way, it makes our definition of routing security satisfiable. As an illustrative example, we considered an authenticated version of the TinyOS beaconing routing protocol, and we showed how an attack against this protocol can be represented in our formal model.

As we mentioned in the Introduction, this paper is a work-in-progress paper. In particular, we have presented neither a new secure routing protocol designed with the help of our formal model, nor a detailed security proof carried out within our model. These are left for future study. We must note, however, that the generality of the simulation paradigm and the fact that we could represent a known attack against the authenticated TinyOS protocol in our model make us confident that we are on the right track.

6 Acknowledgements

The work described in this paper is based on results of IST FP6 STREP UbiSec&Sens (`http://www.ist-ubisecsens.org`). UbiSec&Sens receives research funding from the European Community's Sixth Framework Programme. Apart from this, the European Commission has no responsibility for the content of this paper. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The work presented in this paper has also been partially supported by the Hungarian Scientific Research Fund (contract number T046664). The first author has been further supported by the HSN Lab. The second author has been supported by the Hungarian Ministry of Education (BÖ2003/70).

References

[1] G. Ács, L. Buttyán, and I. Vajda. Provable Security of On-Demand Distance Vector Routing in Wireless Ad Hoc Networks. In *In Proceedings of the Second European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS 2005)*, July 2005.

[2] G. Ács, L. Buttyán, and I. Vajda. Provably Secure On-demand Source Routing in Mobile Ad Hoc Networks. To appear in IEEE Transactions on Mobile Computing.

[3] M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. In *Proceedings of the ACM Symposium on the Theory of Computing*, 1998.

[4] L. Buttyán and I. Vajda. Towards provable security for ad hoc routing protocols. In *Proceedings of the ACM Workshop on Security in Ad Hoc and Sensor Networks (SASN)*, October 2004.

[5] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proc. 42nd IEEE Symposium on Foundations of Computer Science (FOCS)*, 2001.

[6] D. Dolev,and A. C. Yao. On the Security of Public Key Protocols. *IEEE Transactions on Information Theory 29 (2)*, 1983.

[7] IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements. Part 15.4:

Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2003.

[8] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pages 153–181. Kluwer Academic Publisher, 1996.

[9] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, vol. 11, pp. 6-28, 2004.

[10] C. Karlof, D. Wagner. Secure routing in wireless sensor networks: attacks and countermeasures. *Ad Hoc Networks*, Vol. 1, 2003.

[11] Q. Li and J. Aslam and D. Rus. Hierarchical Power-aware Routing in Sensor Networks. In *Proceedings of the DIMACS Workshop on Pervasive Networking*, May, 2001.

[12] J. Marshall. An Analysis of the Secure Routing Protocol for mobile ad hoc network route discovery: using intuitive reasoning and formal verification to identify flaws. MSc thesis, Department of Computer Science, Florida State University, April 2003.

[13] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, February 1999.

[14] A. Perrig, R. Szewczyk, V. Wen, D. Culler, J. D. Tygar. SPINS: Security Protocols for Sensor Networks. In *Wireless Networks Journal (WINE)*, Volume 8, September 2002.

[15] B. Pfitzman and M. Waidner. A model for asynchronous reactive systems and its application to secure message transmission. In *Proc. 22nd IEEE Symposium on Security & Privacy*, 2001.

[16] S. Singh, M. Woo, and C. Raghavendra. Power-Aware Routing in Mobile Ad Hoc Networks. In *Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, Oct. 1998.

[17] W. Xu, W. Trappe, Y. Zhang and T. Wood. The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks. In the *Proceedings of MobiHoc'05*, May 2005.

[18] S. Yang and J. Baras. Modeling vulnerabilities of ad hoc routing protocols. In *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks*, October 2003.