The Security Proof of a Link-state Routing Protocol for Wireless Sensor Networks*

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

Abstract

In this paper, we present a flexible and mathematically rigorous modeling framework for analyzing the security of sensor network routing protocols. Then, we demonstrate the usage of this framework by formally proving that INSENS (Intrusion-Tolerant Routing in Wireless Sensor Networks), which is a secure sensor network routing protocol proposed in the literature independently of our work, can be proven to be secure in our model.

1 Introduction

Most of the sensor network routing protocols proposed in the recent past are subject to various attacks [5]. In order to remedy this situation, some researchers have started to develop secured routing protocols for wireless sensor networks (see e.g., [4, 10]), but provided only an informal security analysis of their protocols. It is well-known, however, that informal reasoning about security is often not reliable enough, as it is quite easy to overlook subtle weaknesses in complex protocols.

In this paper, we propose a mathematically rigorous, yet flexible, modeling framework which supports the reliable security analysis of sensor network routing protocols. This framework extends our prior works [1, 2]. In [2], we proposed a similar framework for ad hoc network routing protocols, and in [1], we adopted that framework for sensor network routing protocols. However, the adversary model in [1] was quite limited and it assumed only an outsider adversary who cannot corrupt legitimate sensor nodes. One of the main contributions of this paper is that we extend the adversary model to insider adversaries who can corrupt some sensor nodes and use the compromised cryptographic material to mount stronger attacks. At the same time, we somewhat simplified the presentation of the framework in this paper, which makes it easier to understand and use it. In addition, another important contribution of this paper is that we also illustrate how our formal framework can be used in practice by proving the security of an existing sensor network routing protocol called INSENS [3]. It is important to note that INSENS was designed by other researchers, independently of our work. During this analysis, we identify a requirement of secure link-state routing protocols that is far more important than it appears at the first sight.

The rest of the paper is organized as follows: In Section 2, we give an overview of the related work. In Section 3, we present our modeling framework, and in Section 4, we demonstrate the usage of the framework by proving the security of INSENS. Finally, in Section 5, we conclude the paper.

2 Related work

Our work is mostly related to [1, 2]. In [2], the authors proposed a formal model based on the simulation paradigm to analyze the security of ad hoc network routing protocols. This simulation-based model was adopted to wireless sensor networks in [1]. The model, in [1], incorporates a new adversary model that is specific to sensor networks, and the authors also modelled the various security objectives in sensor networks in a general manner. However, they came up with neither security proofs nor proof techniques. Moreover, their adversary model is limited in a way that she is assumed not to corrupt legitimate sensor nodes. In this work, we relax this simplifying assumption, and we introduce a more powerful adversary that can control legitimate sensor nodes during a protocol run. In addition, we also demonstrate how our formal technique can be applied to real protocols.

There are some routing protocols proposed for wireless sensor networks with security in mind [10, 4]. In [3, 4], the authors propose an intrusion tolerant routing protocol for wireless sensor networks. INSENS is a centralized link-state routing protocol, where the link-state information do

*The work described in this paper is based on results of IST FP6 STREP UbiSec&Sens (http://www.ist-ubisecsens.org). The work presented in this paper has also been partially supported by the Hungarian Scientific Research Fund and the HSN Lab.

1-4244-1455-5/07/$25.00 ©2007 IEEE

not need to be modified by other nodes during the transit towards the base station, and thus, it implicitly eliminates many potential attacks. Although the authors showed that INSENS [4] successfully mitigate selective forwarding, black hole, and denial-of-service (DoS) attacks, [4, 3] do not contain rigorous security analysis. In Section 4, we will show that INSENS is indeed secure in our model with respect to a security objective specifically tailored for centralized link-state routing protocols in sensor networks.

In [5], the authors informally investigate some attacks against existing sensor network routing protocols. In that paper, routing security is defined implicitly as resistance to these attacks, and the proposed countermeasures are only related to these specific attacks. This informal reasoning is not sufficient to compare the sensor network routing protocols in terms of security, since we do not know what secure sensor network routing exactly means. Moreover, the protocols discussed in [5] has not been designed with security in mind.

In the literature, there are some prior works [6, 9, 7, 8] that also used formal techniques to model the security of multi-hop routing protocols. However, they were mainly proposed for ad hoc network routing, and they either inherently differ from simulation-based models [9, 7, 8], or they are limited to model some protocol specific attacks (like rushing) [6]. In contrast to this, in our work, we are concerned with more general security objectives.

3 Model

Adversary model: Our adversary model is similar to [1] with the exception that when the adversary captures honest sensor nodes in our model, she may be able to compromise their cryptographic secrets (assuming that such secrets are used in the system). Thus, we assume in our model, in contrast to [1], that the adversary can compromise cryptographic material (i.e., our adversary is an *insider adversary* in this sense). Since each adversarial node is assumed to communicate with each other via out-of-band channels, it is also quite natural that all adversarial nodes can use all compromised cryptographic secrets.

In our model, the adversary intends to thwart the primary objectives of routing protocols. Generally, the primary goals of the adversary can be degrading the packet delivery ratio, increasing his control over traffic, increasing network delay, and shortening network lifetime depending on the routing objectives. When attacking protocols, the adversary performs simple message manipulations: injection, deletion, modification, and re-ordering of messages, as well as relaying them without following the routing protocol rules faithfully. Detailed scenarios of performing such message manipulations are described in [1].

Static model: The honest nodes in the network are denoted by $v_0, \ldots, v_k$, where v_0 denotes the base station, and adversarial nodes are denoted by $v_{k+1}, \ldots, v_{k+m}$. The set of all nodes in the network is denoted by V, and the set of adversarial nodes is denoted by V^*, where $|V| = n = m + k + 1$, and $|V^*| = m$.

In order to model the connectivity between the nodes, we introduce a matrix $\mathbf{E}$, called *reachability matrix*, with size $n \times n$. Here, $E_{i,j}$ $(0 \leq i, j \leq n-1)$ represents the energy level needed for v_i to communicate with v_j (i.e., if node v_i uses energy level $E_{i,j}$ to broadcast a message, then v_j also receives the message).

Since adversarial nodes can communicate via out-of-band channels, we merge each adversarial node into a single adversarial node. Accordingly, we model the modified connectivity by matrix $\mathbf{E}^*$, called *reduced reachability matrix*. $\mathbf{E}^*$ can be unambiguously derived from from $\mathbf{E}$ with size $(k+2) \times (k+2)$ in the following way. For all i, j $(0 \leq i, j \leq k)$, $E^*_{i,j}$ is identical to $E_{i,j}$. For an honest node v_ℓ $(0 \leq \ell \leq k)$, $E_{\ell,k+1}$ represents the minimal energy level that is needed for v_ℓ to communicate with at least one adversarial node. Similarly, $E_{k+1,\ell}$ represents the minimal energy level that is needed for the adversary to communicate with v_ℓ (i.e., there exists at least one adversarial node that can communicate with v_ℓ using energy level $E_{k+1,\ell}$).

Finally, a *cost function* $\mathcal{C} : V \rightarrow \mathbb{R}$ assigns a cost value to each node in the network (e.g., the remaining energy in the battery, or constant 1 to each node in order to represent hop-count, etc.) that could influence the routing decisions.

The *configuration* of a network is a quardlet $conf = (V, V^*, \mathbf{E}, \mathcal{C})$, where V and V^* are the set of honest nodes and the set of adversarial nodes, resp., and $\mathbf{E}$ is the reachability matrix.

Security objective function: In order to model different security objectives in a general manner, we introduce the security objective function [1]. We represent the output of a routing protocol, which is the ensemble of the routing entries of the honest nodes, with a given configuration *conf* by a matrix $\mathbf{T}^{conf}$ with size $(k+1) \times (k+2)$:

- for $0 \leq i, j \leq k$, $T^{conf}_{i,j} = 1$, if honest node v_i forwards every data message to another honest node v_j in order to deliver the message to the base station, otherwise $T^{conf}_{i,j} = 0$,
- for $0 \leq i \leq k$ and $j = k+1$, $T^{conf}_{i,j} = 1$, if honest node v_i forwards every data message to an adversarial node in order to deliver the message to the base station (i.e., v_i sets a corrupt node as a next-hop towards the base station), otherwise $T^{conf}_{i,j} = 0$.

Actually, $\mathbf{T}^{conf}$ is a random variable due to the randomness in sensor readings, processing and transmission time, etc. In the sequel, we also refer to $\mathbf{T}^{conf}$ as the *routing topology* of configuration $conf$, and we will omit the index $conf$ when the configuration can be unambiguously determined in a given context. The security objective function $\mathcal{F} : \mathbb{G} \times \mathbb{T} \to \{0, 1\}$ is a binary function, where $\mathbb{T}$ denotes the set of routing topologies of all configurations, and $\mathbb{G}$ denotes the set of all configurations. This function intends to distinguish "attacked" (incorrect) topologies from "non-attacked" (correct) topologies based on a well-defined security objective.

For example, let us consider routing protocols that build a routing tree, where the root is the base station. We can construct a security objective function based on network lifetime as follows:

$$\mathcal{F}(conf,\mathbf{T}^{conf})=\begin{cases}1, & \frac{1}{k}\sum_{i=0}^{k}\sum_{j=0}^{k+1}T_{i,j}\cdot {E^*_{i,j}}^{\alpha}\cdot \mathcal{C}(v_j)^{\beta}\leq c\\ 0, & \text{otherwise}\end{cases}$$

where α and β are tunable weighting factors (i.e., protocol parameters), and $\mathcal{C}$ represents the remaining energy level. $\mathcal{F}$ returns 1 for all topologies, where the average cost of the entries set by honest nodes is upper bounded by a constant number c. Since $\mathbf{T}^{conf}$ is a random variable, the output of $\mathcal{F}$ is a random variable too.

In the rest of the paper, we assume that $\mathcal{F}$ returns 1 if the routing topology is correct. Otherwise, it returns 0.

Dynamic model: The dynamic model is similar to [1, 2]. However, our model deviates from these works in the sense that we do not distinguish a real-world model and an ideal-world model as usual in the simulation paradigm, but for the simplicity of the presentation, we define a single model that represents the real operation of the network. The security objective function is applied to the output of this model (i.e., the resulting routing topology) in order to decide whether the protocol functions correctly or not.

We denote the output by $Out^{\mathcal{F}}_{conf,\mathcal{A}}(r)$, where r is the random input of the model. In addition, $Out^{\mathcal{F}}_{conf,\mathcal{A}}$ will denote the random variable describing $Out^{\mathcal{F}}_{conf,\mathcal{A}}(r)$ when r is chosen uniformly at random.

Definition of secure routing: We 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 MAC, digital signature etc.). Based on the model described in the previous subsections, we define routing security as follows:

Definition 1 *A routing protocol is secure with security objective function $\mathcal{F}$, if for any configuration conf and any adversary $\mathcal{A}$, the probability that $Out^{\mathcal{F}}_{conf,\mathcal{A}}$ equals to zero is a negligible function of κ.*[1]

More intuitively, if a routing protocol is secure, then any system using this routing protocol may not 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 MAC) with a very small probability depending on the value of κ.

4 Security of INSENS

4.1 Operation of INSENS

In this subsection, we describe the operation of INSENS (for more detailed description, see [3]). In this paper, we are only concerned with the topology (route) discovery mechanism of INSENS and not with the data forwarding mechanism.

Calculation of neighborlist: The base station initiates the routing topology construction by flooding the network with a route request message, which has the following format:

$$v_0 \rightarrow * : (\mathsf{REQ}, \mathsf{hash}, [v_0])$$

where REQ is a constant message type identifier, hash is the next element of the hash chain in reversed direction, and v_0 identifies the base station. The hash chain mechanism is intended to provide authenticity and some defense against DoS attacks. Each node constructs its own neighborlist by overhearing the request messages sent by its neighbors.

Every subsequent node v_{ℓ_i} receiving request

$$(\mathsf{REQ}, \mathsf{hash}, [v_0, v_{\ell_1}, \ldots, v_{\ell_{i-1}}], \mathsf{MAC}^{\mathsf{REQ}}_{v_{\ell_{i-1}}})$$

verifies the correctness of hash and checks whether it is the first request containing hash. If it is the first one, then v_{ℓ_i} re-broadcasts the modified request, and stores $\mathsf{MAC}^{\mathsf{REQ}}_{v_{\ell_{i-1}}}$ in conjunction with $\mathcal{L}(v_{\ell_{i-1}})$ locally. Before re-broadcasting, v_{ℓ_i} replaces $\mathsf{MAC}^{\mathsf{REQ}}_{v_{\ell_{i-1}}}$ in the request to $\mathsf{MAC}^{\mathsf{REQ}}_{v_{\ell_i}}$, which is the MAC generated by v_{ℓ_i} on list $[v_0, \ldots, v_{\ell_{i-1}}, v_{\ell_i}]$, REQ, and hash using the symmetric key shared with v_0. Finally, v_{ℓ_i} re-broadcasts the following request:

$$v_{\ell_i} \rightarrow * : (\mathsf{REQ}, \mathsf{hash}, [v_0, \ldots, v_{\ell_{i-1}}, v_{\ell_i}], \mathsf{MAC}^{\mathsf{REQ}}_{v_{\ell_i}})$$

Forwarding neighborlist towards the base station: If a node v_{ℓ_x} does not receive further request messages for a

[1] a function $\mu(x) : \mathbb{N} \to \mathbb{R}$ is negligible, if for every positive integer c and all sufficiently large x's (i.e., there exists an $N_c > 0$ for all $x > N_c$), $\mu(x) \leq x^{-c}$

specified time, v_{ℓ_x} sends the following message to $v_{\ell_{x-1}}$ from which it received the first valid request:

$$
\begin{aligned}
&v_{\ell_x} \rightarrow v_{\ell_{x-1}} : \\
&\qquad (\mathsf{NLIST}, \mathsf{hash}, \mathsf{MAC}^{\mathsf{REQ}}_{v_{\ell_{x-1}}}, v_{\ell_x}, \\
&\qquad\quad \mathsf{Enc}_{v_{\ell_x}}(path_{v_{\ell_x}}, neighborlist_{v_{\ell_x}}), \mathsf{MAC}^{\mathsf{NLIST}}_{v_{\ell_x}})
\end{aligned}
$$

where the elements of the message are as follows: NLIST is a constant message type identifier; hash is the hash value of the corresponding request message; $\mathsf{MAC}^{\mathsf{REQ}}_{v_{\ell_{x-1}}}$ is the MAC, called parent MAC[2], of $v_{\ell_{x-1}}$ sent in the corresponding request; v_{ℓ_x} is the identifier of the message originator; $\mathsf{Enc}_{v_{\ell_x}}(path_{v_{\ell_x}}, neighborlist_{v_{\ell_x}})$ is the neighborhood information and the path information of v_{ℓ_x} encrypted by the symmetric key shared with the base station; $neighborlist_{v_{\ell_x}}$ contains the identifiers of each neighboring node *and* their corresponding MACs received in Phase 1; $path_{v_{\ell_x}}$ is $[v_{\ell_x}, \ldots, v_{\ell_1}, v_0, \mathsf{MAC}^{\mathsf{REQ}}_{v_{\ell_x}}]$, which is the reverse of the path received in the corresponding request message including the MAC of node v_x; and finally $\mathsf{MAC}^{\mathsf{NLIST}}_{v_{\ell_x}}$ is the MAC computed by node v_{ℓ_x} on NLIST, hash, $path_{v_{\ell_x}}$, and $neighborlist_{v_{\ell_x}}$.

A node receiving the reply message first checks if the node is the parent of the sender (i.e., $\mathsf{MAC}^{\mathsf{REQ}}_{v_{\ell_{x-1}}}$ message equals to its own MAC that has been broadcast with request containing hash). Then, the node replaces the parent MAC in the message to its own parent MAC that is stored in Phase 1. In this way, the reply message propagates back to the base station. Upon the reception of a reply message

$$
(\mathsf{NLIST}, \mathsf{hash}, v_{\ell_x}, \mathsf{Enc}_{v_{\ell_x}}(path_{v_{\ell_x}}, neighborlist_{v_{\ell_x}}), \mathsf{MAC}^{\mathsf{NLIST}}_{v_{\ell_x}})
$$

the base station checks whether all the MACs are correct, after decrypting $\mathsf{Enc}_{v_{\ell_x}}(path_{v_{\ell_x}}, neighborlist_{v_{\ell_x}})$[3]. If all verifications are successful, the base station computes the forwarding table for each node using a global centralized algorithm detailed in [3].

Distributing forwarding tables: The forwarding tables are propagated to respective nodes in a breadth-first manner; first, the immediate neighbors of the base station receive their forwarding tables directly from the base station. Afterwards, these one-hop neighbors forward the forwarding tables of the two-hop neighbors of the base station based on their forwarding tables, and so on. In particular, the base station first sends the forwarding table of v_{ℓ_1}:

$$
v_0 \rightarrow v_{\ell_1} : (\mathsf{FTABLE}, v_{\ell_1}, \mathsf{hash}, \mathsf{Enc}_{v_{\ell_1}}(ftable_{v_{\ell_1}}), \mathsf{MAC}^{\mathsf{FTABLE}}_{v_{\ell_1}})
$$

where FTABLE is a constant message type identifier, $\mathsf{Enc}_{v_{\ell_1}}(ftable_{v_{\ell_1}})$ is the encrypted form of the forwarding table of v_{ℓ_1}, and $\mathsf{MAC}^{\mathsf{FTABLE}}_{v_{\ell_1}}$ is the MAC generated by v_0 on the complete message. Upon the reception of this message, v_{ℓ_1} sets its forwarding rules according to $ftable_{v_{\ell_1}}$, if $\mathsf{MAC}^{\mathsf{FTABLE}}_{v_{\ell_1}}$ is correct.

4.2 Security proof

In this subsection we show that INSENS described in Section 4.1 is secure in our model. We show that the protocol has the following properties:

1. If an honest sensor node v_i $(1 \leq i \leq k)$ sets $v_j \in V$ $(0 \leq j \leq n-1)$ as its parent node for data forwarding, then the base station has indeed computed v_j as the parent node for v_i.

2. If the base station is aware of the fact that node v_j is a neighbor of node v_i, then node v_i can reach node v_j by either a direct contact, or an adversarial relaying (one can also imagine the adversarial relaying as a wormhole between some honest nodes).

Intuitively, if INSENS has these two properties, then it is ensured that each honest node has a *neighboring* parent node that is computed by the base station. Moreover, it is also guaranteed that this computation performed by the base station is based on, perhaps incomplete (the adversary can always drop routing messages containing neighborlists, which we are unable to defend against), but correct neighborhood information. In fact, this is a general security objective of every kind of link-state routing protocol for sensor networks.

In order to formalize the above security objective, we introduce a matrix function $\mathcal{G}$. $\mathcal{G}$ models the centralized construction of the topology performed by the base station, where the argument of $\mathcal{G}$ with size $(k+2) \times (k+2)$, denoted by $\mathbf{N}$, describes the neighborhood relations among the sensor nodes that is believed by the base station to be correct (i.e., $N_{i,j} = 1$ if the base station believes that v_i is a neighbor of v_j, otherwise $N_{i,j} = 0$). The output of $\mathcal{G}$ is the ensemble of the routing entries (the routing topology) that should be set by each node.

Now, we prove that INSENS is secure with respect to the aforementioned security objective.

Theorem 1 *Let us consider the following security objective function:*

$$
\mathcal{F}(conf, \mathbf{T}) = \begin{cases} 1, & \textit{there exists } \mathbf{E}' \textit{ such that for all } i,j \textit{ it holds that if } T_{i,j} = 1, \textit{ then } \mathcal{G}(\mathbf{E}')_{i,j} = 1 \\ 0, & \textit{otherwise} \end{cases}
$$

[2] In this context, parent node is the next-hop that forwards neighborhood information, and not measured data, towards the base station.

[3] Actually, the MACs in the $neighborlist_{v_{\ell_x}}$ can only be checked when the NLIST messages of the corresponding nodes in $neighborlist_{v_{\ell_x}}$ are also received.

where $\mathbf{E}'$ with size $(k+2) \times (k+2)$ is derived from $\mathbf{E}^*$, such that $E'_{i,j} = 0$, if $E^*_{i,j} = \infty$, and $E^*_{i,k+1} = \infty$ or $E^*_{k+1,j} = \infty$[4]. INSENS is secure with respect to $\mathcal{F}$, if the MAC scheme is secure against existential forgery, and the symmetric encryption scheme is secure against plaintext recovery attack.

Proof We show that for any adversary $\mathcal{A}$ and any configuration *conf*, $\mathcal{F}(conf, \mathbf{T}) = 0$ only with probability that is a negligible function of κ_1 and κ_2, where κ_1, κ_2 are the security parameters of the employed MAC and encryption schemes, resp. In other words, the success probability of any adversary is a negligible function of κ_1 and κ_2.

From the definition of $\mathcal{F}$, $\mathcal{F}(conf, \mathbf{T}) = 0$ if there exist i, j $(1 \leq i \leq k, 0 \leq j \leq k+1)$ such that $T_{i,j} = 1$ and there does not exist any $\mathbf{E}'$, derived from $\mathbf{E}^*$, such that $\mathcal{G}(\mathbf{E}')_{i,j} = 1$. This can have two reasons as follows: (i) node v_i received incorrect routing topology information, or (ii) the base station received incorrect neighborhood information. According to this, we introduce the following events:

(i) $\mathsf{C}_1^{i,j}$ denotes the event that $T_{i,j} = 1$, but $\mathcal{G}(\mathbf{N})_{i,j} = 0$,

(ii) $\mathsf{C}_2^{i,j}$ denotes the event that $T_{i,j} = 1$, $\mathcal{G}(\mathbf{N})_{i,j} = 1$, and $N_{i,j} = 1$, but $E^*_{i,j} = \infty$ as well as $E^*_{i,k+1} = \infty$ or $E^*_{k+1,j} = \infty$.

We recall that $\mathbf{N}$ describes the neighborhood relations among the sensor nodes, which is believed by the base station to be correct. Clearly, the following upper estimation holds for the success probability of the adversary denoted by $P^{\mathcal{A}}$:

$$P^{\mathcal{A}} \leq \sum_{\forall i,j: i \neq j, i \neq 0} \mathbf{P}\left(\mathsf{C}_1^{i,j}\right) + \sum_{\forall i,j: i \neq j, i \neq 0} \mathbf{P}\left(\mathsf{C}_2^{i,j}\right)$$

We show that $\mathbf{P}\left(\mathsf{C}_1^{i,j}\right)$ is a negligible function of κ_1, and $\mathbf{P}\left(\mathsf{C}_2^{i,j}\right)$ is a negligible function of κ_1 and κ_2 for all i, j. This implies that $P^{\mathcal{A}}$ is also a negligible function of κ_1 and κ_2 that concludes the theorem.

Negligibility of $\mathbf{P}\left(\mathsf{C}_1^{i,j}\right)$: If $\mathsf{C}_1^{i,j}$ occurs, then M_i receives an FTABLE message, which contains the routing information of node v_i:

$$(\mathsf{FTABLE}, v_i, \mathsf{hash}, \mathsf{Enc}_{v_i}(ftable'_{v_i}), \mathsf{MAC}'^{\mathsf{FTABLE}}_{v_i})$$

v_i infers from $ftable'_{v_i}$ that $T_{i,j} = 1$, since $\mathsf{MAC}'^{\mathsf{FTABLE}}_{v_i}$ is a correct MAC. We show that it is only possible if $\mathsf{MAC}'^{\mathsf{FTABLE}}_{v_i}$ is a successfully forged MAC by A.

Let us assume that A cannot forge $\mathsf{MAC}'^{\mathsf{FTABLE}}_{v_i}$. Hence, M_0 is the only machine who generates $\mathsf{MAC}'^{\mathsf{FTABLE}}_{v_i}$. However, M_0 generates $\mathsf{MAC}'^{\mathsf{FTABLE}}_{v_i}$ only if $[\mathcal{G}(N)]_{i,j} = 1$, which is a contradiction.

Consequently, $\mathsf{C}_1^{i,j}$ occurs for any i, j, if the adversary $\mathcal{A}$ successfully forges a MAC. However, the probability of this event is a negligible function of κ_1 assuming that $\mathcal{A}$ runs in polynomial time.

Negligibility of $\mathbf{P}\left(\mathsf{C}_2^{i,j}\right)$: If $\mathsf{C}_2^{i,j}$ occurs, then M_0 receives an NLIST message, which contains the neighborhood information of node v_j:

$$(\mathsf{NLIST}, \mathsf{hash}, v_j, \mathsf{Enc}_{v_j}(path_{v_j}, neighborlist'_{v_j}), \mathsf{MAC}'^{\mathsf{NLIST}}_{v_j})$$

v_0 infers from $neighborlist'_{v_j}$ that $N_{i,j} = 1$, since $\mathsf{MAC}'^{\mathsf{NLIST}}_{v_j}$ is a correct MAC. We show that it is only possible if at least one of the following conditions holds:

1. $\mathsf{MAC}'^{\mathsf{NLIST}}_{v_j}$ is a successfully forged MAC by A, if v_j is an honest node.

2. There exists a node v_t $(1 \leq t \leq k)$, for which $E^*_{i,t} < \infty$ and A successfully recovered the plaintext from $\mathsf{Enc}_{v_t}(path_{v_t}, neighborlist_{v_t})$ that is sent in the corresponding NLIST message by v_t.

3. $\mathsf{MAC}'^{\mathsf{REQ}}_{v_i}$ that is received by v_j is a successfully forged MAC by A.

Let us assume that *none* of the above conditions hold. Two main cases can be distinguished: (i) v_j is an honest node, or (ii) v_j is an adversarial node.

(i) Based on the argument of the negligibility of $\mathsf{C}_1^{i,j}$, we know that $\mathsf{MAC}'^{\mathsf{NLIST}}_{v_j}$ can only be generated by M_j. Thus, M_j received a REQ message denoted by

$$msg' = (\mathsf{REQ}, \mathsf{hash}, [v_0, \ldots, v_i], \mathsf{MAC}'^{\mathsf{REQ}}_{v_i})$$

We know that msg' is never relayed by machines $M_0, \ldots, M_{i-1}, M_{i+1}, \ldots, M_k$, since these machines never send any REQ messages containing a path where the last element is v_i (such as path $[v_0, \ldots, v_i]$ in msg'). Therefore, M_j receives msg' from A implying that $E^*_{k+1,j} < \infty$.

Since v_i is not an adversarial node, $\mathsf{MAC}'^{\mathsf{REQ}}_{v_i}$ cannot be generated by machines $M_0, \ldots, M_{i-1}, M_{i+1}, \ldots, M_k, A$. Therefore, only M_i can generate $\mathsf{MAC}'^{\mathsf{REQ}}_{v_i}$. We know that msg' cannot be sent to M_j by M_i, since $E_{i,j} = \infty$. We will show that $E^*_{i,k+1} < \infty$, which is a contradiction.

First, let us assume that $E^*_{i,k+1} = \infty$. In order to construct msg', A can only infer $\mathsf{MAC}'^{\mathsf{REQ}}_{v_i}$

[4] The rationale behind the definition of $\mathbf{E}'$ is that the adversary can always drop messages that should be tolerated. However, we can defend against illegal injection and modification of messages by using appropriate cryptographic primitives.

from the messages sent by the neighbors v_t of v_i, since only honest nodes v_t can be reached by v_i, and these nodes only relay $\mathsf{MAC}'^{\mathsf{REQ}}_{v_i}$ in an encrypted form. In that case, $\mathsf{MAC}'^{\mathsf{REQ}}_{v_i}$ must be inferred from $\mathsf{Enc}_{v_t}(path_{v_{\ell_t}}, neighborlist_{v_t})$, which contradicts to our assumption. Therefore, $E^*_{i,k+1} < \infty$.

(ii) Let us assume that $E^*_{i,j} = \infty$, where $j = k + 1$. Similar to case (i), A can only infer $\mathsf{MAC}'^{\mathsf{REQ}}_{v_i}$ from the messages sent by the neighbors of v_i, as A is unable to forge $\mathsf{MAC}'^{\mathsf{REQ}}_{v_i}$. Thus, A must recover $\mathsf{MAC}'^{\mathsf{REQ}}_{v_i}$ from encrypted neighborlists. However, by assumption, the adversary cannot do this. This means that $E^*_{i,j} < \infty$, which is a contradiction again.

Consequently, $\mathsf{C}_2^{i,j}$ can only occur for any i, j, if at least one of the above conditions is true. This implies that the adversary $\mathcal{A}$ is able to forge a MAC, or $\mathcal{A}$ can recover the plaintext from a ciphertext. However, the probability of this event is a negligible function of κ_1 and κ_2 assuming that $\mathcal{A}$ runs in polynomial time. ■

5 Conclusion

In this paper, we proposed a formal framework to analyze the security of routing protocols in wireless sensor networks. This model encompasses a strong adversary model, which may also participate in the routing process as a legitimate node. We modelled the security objectives in a very general manner, and thus, various sensor network routing protocols can be analyzed in our model in a flexible way. After describing our model, we demonstrated this technique on a real example: we proved that INSENS, which is a secure sensor network routing protocol, is indeed secure in our model.

We recall that the proof is strongly based on the assumption that the encryption scheme is secure against plaintext recovery attack. The encryption of neighborlists used in INSENS is crucial; apart from providing confidentiality for the neighborhood relations, the encryption of neighborlists prevents the adversary to impersonate honest nodes that are not covered by the transmission range of any adversarial nodes. For instance, if the neighborlists were not encrypted, an intermediate adversarial node could easily retrieve the identities and corresponding $\mathsf{MAC}^{\mathsf{REQ}}$s from NLIST messages, and then she could re-broadcast fabricated REQ messages. Note that the adversary is not required to reach the impersonated node directly. Apparently, this would also violate our security objective detailed in Subsection 4.2, as the adversary could cause the base station to consider false neighborhood relations. Furthermore, as $\mathsf{MAC}^{\mathsf{REQ}}$s are correct, it can happen that neither the neighbors of the adversary nor the base station could detect the misdeed. This attack scenario was not described in [3], where the authors used informal reasoning to prove the security of INSENS.

In contrast to this, our formal security analysis would reveal such flaw in a routing protocol: if encryption had not been employed, we could not have claimed in the proof that the adversary can retrieve the $\mathsf{MAC}^{\mathsf{REQ}}$ of a non-neighboring node only from the encrypted neighborlist of other nodes. Therefore, our formal analysis lead us to the following observation: in case of link-state routing, all local neighborhood (routing) information that is needed by remote nodes to authenticate neighborhood relations must be transferred in an encrypted form.

References

[1] G. Ács, L. Buttyán, and I. Vajda. Modelling Adversaries and Security Objectives for Routing Protocols in Wireless Sensor Networks. In *Proceedings of ACM SASN*, Oct. 2006.

[2] G. Ács, L. Buttyán, and I. Vajda. Provably Secure On-demand Source Routing in Mobile Ad Hoc Networks. In *IEEE Transactions on Mobile Computing*, Vol. 5, No. 11, November 2006.

[3] J. Deng, R. Han, and S. Mishra. INSENS: Intrusion-Tolerant Routing in Wireless Sensor Sensor Networks. *Technical Report CU-CS-939-02*, Department of Computer Science, University of Colorado, November 2002.

[4] J. Deng, R. Han, and S. Mishra. A performance evaluation of intrusion-tolerant routing in wireless sensor networks. In *IEEE Workshop on Information Processing in Sensor Networks (IPSN)*, pages 349-364, Apr. 2003.

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

[6] J. Kong, X. Hong, and M. Gerla. Modeling Ad-hoc Rushing Attack in a Negligibility-based Security Framework. In *Proceedings of the 5th ACM Workshop on Wireless Security (WiSe)*, pp. 55-64, 2006.

[7] 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.

[8] P. Papadimitratos, Z.J. Haas, and J.-P. Hubaux. How to Specify and How to Prove Correctness of Secure Routing Protocols for MANET. In *Proceedings of IEEE CS BroadNets 2006*, San Jose, CA, October 2006.

[9] 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.

[10] A. D. Wood, L. Fang, J. A. Stankovic, and T. He. SIGF: A family of configurable, secure routing protocols for wireless sensor networks. In *Proceedings of ACM SASN*, Oct. 2006.