Low-Cost Secret-Sharing in Sensor Networks

Marin Bertier, Achour Mostefaoui, Gilles Tredan

2010 IEEE 12th International Symposium on High Assurance Systems Engineering

Low-Cost Secret-Sharing in Sensor Networks

Marin BERTIER Achour MOSTEFAOUI
IRISA, Université de Rennes 1
35042 Rennes, France
firstnamne.lastname@irisa.fr

Gilles TRÉDAN
Deutsche Telekom / TU Berlin
D-10587 Berlin, Germany
Email: gilles@net.t-labs.tu-berlin.de

***Abstract*—Radio waves are the medium used by sensors to communicate and exchange data. The unconstrained accessibility to any information carried over this medium is a security issue in many sensor-based applications.**

Ensuring protected wireless communications is a problem that has received a lot of attention in the context of ad hoc networks. However, due to hardware constraints of sensors along with multi-hop communication, most of these solutions turn out to be useless for sensor networks.

This paper provides basic building blocks to establish secure communication by exchanging secret keys between neighbor nodes without any use of cryptography methods allowing an gain in efficiency. This paper also proposes a second algorithm that extends the secret key establishment to nodes that are not direct neighbors. Among the interesting features of the proposed algorithms we can note a low overhead and the absence of initial configuration.

Keywords: Distributed algorithm, Malicious behavior, Reliability, Resiliency, Sensor network, Wireless communication.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are large networks composed by tiny devices with very restricted resources in terms of computational power, memory and above all energy. The fact that sensors communicate using a radio technology medium, allows any entity located within the communication range of an emitting sensor to receive sent messages and hence compromise the *privacy* of data exchanged. As sensors can be deployed in a large area, and as the communication range is small, sensors have to use multi-hop communication to reach the sink. In consequence, the number of sensors which relay the data is high (routing sensors). This allows malicious nodes to compromise the *integrity* by modifying the data they relay or by lying about its origin.

Several approaches have been proposed to protect communication between two neighbors based on cryptography. Due to the limited resources of sensors, the use of asymmetric cryptography was considered very costly [2]. Recently, the use of elliptic curve cryptography based on exponentiation reduced drastically this cost however it still needs a public key infrastructure. A similar problem holds for the use of symmetric cryptography which requires an initial configuration [3], [5], [9], [12]. To be able to communicate, any pair of neighbor sensors has to share an initial key. A straightforward solution consists in assigning a personal key to each node and then distributing the key of each sensor to its neighbors. Because the network configuration has to be known, this kind of solution is very constraining or simply impossible if the sensors are disseminated by an airplane or in a hostile area. Moreover, this entails an ad hoc parametrization of all the sensors, hence the impossibility to automate the programming of the sensors. To avoid this constraint, solutions based on the birthday paradox have been proposed [11]. This allows the use of symmetric cryptography without any a priori knowledge of the network topology but imposes an expensive initial step after the network bootstrap in order to localize the keys and then rearrange them. Finally, existing approaches require a tedious configuration and / or an expensive initial step to allow the establishment of a protected communication between two nodes. Moreover, as a key is obtained by configuration, it cannot be renewed as easily as it can be revealed by capture-attacks.

In this paper, we propose a new approach to establish protected communications in sensor networks without any initial configuration (all the nodes can have the exact same initial configuration except their own id) and the protection can be renewed at any point of the execution. Our approach allows to share a secret between two neighbors based only on the geographically uniform distribution of the nodes. More precisely, it relies on the observation that many pairs of neighboring nodes have a unique subset of common neighbors. This allows to establish one-hop protected communications. We then show that it is possible to rely on these protected neighboring communications to establish protected multi-hop communications. If the geographical structure of the network is imposed, it is not always possible to protect the communication between any pair of neighbors, we provide a probabilistic security scheme comparable to [11]. However, a simulation study shows that, in an area where nodes are uniformly distributed, more than 85% of nodes are able to establish a protected communication with their neighbors. We also show in Section V that for some structures, the establishment of protected communications is always possible.

The paper is composed of 6 sections. Section II presents

1530-2059/10 $26.00 © 2010 IEEE
DOI 10.1109/HASE.2010.16

the system model with its requirements in terms of system configuration and communication. It also introduces some notions used in the rest of the paper. Section III describes our approach. Namely, how to obtain a secret key shared by two nodes and how to use this secret key to protect multi-hop communication. Section IV presents several simulation results that illustrate the conditions required to establish protected communication. Section V discusses the possibility to relax some assumptions, presents the impact on the system behavior and proposes two ad hoc network structures that make possible protected communication all over the network. Finally, Section VI concludes the paper.

II. System Model

The system model consists of a finite set V of sensors (the size of which is a priori unknown). The system is patterned after the model given in [8].

Communication model: A sensor is equipped with a communication device based on radio technology that allows it to send and receive messages within a certain range (we can say, without loss of generality that communication range is 1). Our algorithm only requires symmetric communications, but for the sake of clarity, we model radio range by a circle. The multi-hop communication system can be represented by an undirected graph $G = (V, E)$ where V is the set of sensors and E the set of edges. There is an edge between sensor u and sensor v if u and v can directly communicate (they are within mutual transmission range). We say that there is one hop between sensor u and sensor v if edge $(u, v) \in E$.

For the sake of presentation, we will only consider the case of Unit disk graphs [4]. This means that G has an embedding such that the Euclidean distance between two sensors is smaller than 1 if, and only if, they are connected by an edge. However, note that our only requirement, for the moment, is that communications have to be symmetric. The symmetry of communication will be relaxed in the Discussion section.

In our model, we consider that sensors are static and have no notion of distance and positioning (e.g. no GPS). The only thing sensors can learn about the topology and the structure of the network is what they obtain by exchanging messages with their immediate neighbors (sensors that are at 1-hop distance). We assume that the communication graph is connected.

Timing model: The system has a synchrony requirement. In order to learn its neighborhood, a sensor sends a message (that will be received by its 1-hop neighbors) and then waits for a response from each of them. As it does not know how many neighbors there are, all it can do is to wait for the responses to arrive. In order to be able to stop this step and then consider that it has got a response from all its neighbors, communication delays need to be bounded.

Adversity model: The basic fault model considered in sensor networks is the crash failure model. It considers that a sensor can only fail by crashing. But sensors may also fail by deviating from their specification. The causes of such misbehavior are twofold: either accidental or intentional.

In fact, when a sensor approaches energy shortage, or due to a misconception or a damaged component, its behavior can deviate from its specification [6]. This means that such a sensor can send messages with wrong values and alter messages it relays. Another source of arbitrary sensor behavior may be an external adversary whose goal is either to make the system behave wrongly, or to collect information and send it to a wrong destination, or to prevent other sensors from communicating by jamming communication links, etc. For such a purpose, the adversary can either add its own sensors that will interfere with the sensors of the network, or hack some of the sensors of the considered network, *e.g.* by capture-attacks [10]. Defining the power of the adversary means also to have an assumption on whether "hacked" sensors know each other and whether they cooperate in their nasty job. In the literature, nodes that may deviate from their specification are called Byzantine nodes [7].

The classical Byzantine model supposes the worst scenario where the adversary has unbounded power. Given the open nature of radio communication, and the resource-constrained nature of sensors, this model is not suited for sensors networks:

- If malicious nodes have no energy constraints, they can jam communication in the network [6] and prevent communication between sensors.
- If their computation power is not bound, the use of cryptography is useless.
- If the number of malicious nodes is high, they can disconnect the network by refusing to forward messages.

In summary, we have to propose an adversary which can be managed, but powerful enough to cope with real attacks against the network.

The interested reader can find in [1] information about the definition of the adversary when dealing with Byzantine behaviors. In this paper, the adversary model we consider is the following. We assume that part of the sensors can exhibit a Byzantine behavior (it does not matter whether these Byzantine sensors are hacked sensors or added sensors). However, we consider that Byzantine sensors do not cooperate together, cannot jam communication, and their computation power is bounded.

We also assume that radio medium is the only way for sensors (Byzantine or not) to communicate. Concerning the communication graph, we consider that the graph G reduced to the non malicious sensors remains connected.

Problem: In this paper, we exclude the use of asymmetric cryptography (costly) or an initial distribution of keys, as in such cases the problem of protected communication becomes trivial. We want to provide each pair of neighbor

sensors with a secret shared key such that no other sensor is able to decipher information ciphered with this key. The use of secret shared keys falls out of this paper's scope. However, once a secret key is established between two sensors, the ciphering can consist, for example, in a simple xor of the message one of the two sensors wants to send with the secret key. Of course, the unciphering will then also consist in a xor operation between the received information and the same key (the operation is symmetric)[1].

The shared secret key problem (SSK problem for short) can be stated informally as follows. Sensors start from a state where the only parameter that distinguishes them is their id. We want to establish protected links between 1-hop distance pairs of sensors. Recall that a sent message is received by all the sensors that are at 1-hop distance. In other words, we want to provide each pair of sensors at 1-hop distance with a secret shared key. Such a link is called a *protected link*.

More formally, each sensor that participates in the shared secret key problem gets as a result a set of keys that satisfy the following three properties:

- Completeness: Each node will get a secret key for each of its 1-hop neighbors.
- Symmetry: The key obtained by a node u to communicate with a neighboring node v is the same as the key obtained by node v to communicate with node u.
- Privacy: The key obtained by two 1-hop neighbors is known only to them.

For such a purpose, nodes will execute an algorithm the output of which is a set of keys. Each node gets as many keys as the number of 1-hop neighbors it has.

III. Proposed Approach

A. Establishing 1-Hop Secret Keys

A node has a fixed set of neighbors. This set depends on its physical location[2]. The objective of this section is to propose an algorithm that solves the SSK problem presented in the previous section, namely, the establishment of 1-hop secret keys.

For the sake of simplicity, we will follow the secret key establishment through an example. Figure 1 represents part of a sensor network composed of a set of nodes $\{A, B, C, D, E\}$. Dotted circles represent the range of nodes C and D, and gray lines symbolise an edge in the communication graph. These nodes only differ by their id. We want to allow node A and node B to get a common secret key. The different nodes around (nodes C, D and E) should not be able to decipher the information exchanged between A and B.

To that end, each node first builds a set containing it's neighbors id (including itself). In Figure 1, the edges

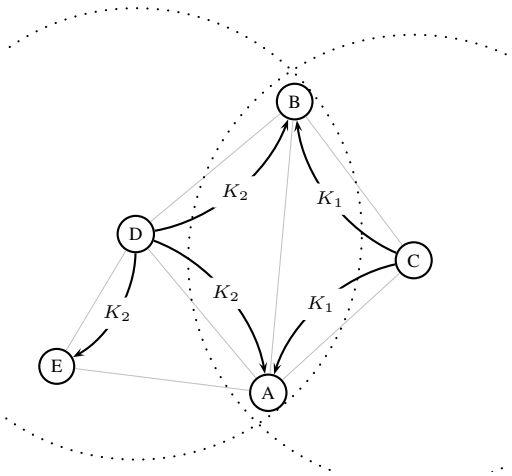


Figure 1. Example of secret establishment

between sensors depict which sensors receive the basic keys sent by nodes C, D. We can observe that the only neighbors nodes A and B have in common are nodes C and D. Note that A and B are also common neighbors themselves. Node E does not belong to this set as it is not an immediate neighbor of node B. The idea is: each common neighbor (C and D) provides one part of the key that will protect the link (A, B). Since only A and B can hear both C and D, they are the only nodes able to compute a key from the two received keyparts.

More formally, let V_i be the set of neighbors of a node i (including i itself). We define the *common neighbors set* of two nodes i and j as $CN_{ij} = V_i \cap V_j$. In order to satisfy the Privacy property of the SSK problem, the neighborhood of A and B has to satisfy some conditions.

- A and B have at least two common neighbors namely A and B. As the SSK problem consists in building protected links between 1-hop distance pairs of sensors A is a neighbor of B and vice versa. As by definition, each node is put in its set of neighbors, the intersection of the neighbors of any couple of neighbor node contains at least the two considered nodes plus eventually other nodes (here, C and D in addition to A and B), each one of the common neighbors providing then part of the key.
- A common neighbor (excluding A and B as they need to be aware of the secret) should not be able to all hear other common neighbors (i.e. be 1-hop neighbor with them), otherwise C would know D's keypart and thus the whole key.

We can formalize these two observations under the condition:

$$COND_{ij} : \cap_{k \in CN_{ij}} V_k = \{i, j\}$$

[1]It is possible to achieve *one-time-password* cryptography by sending a new key each time the previous key has been used, ensuring by this way an unbreakable ciphering.

[2]We consider fixed sensors. They have no way to change their location.

In our example, we have

$$\begin{aligned} COND_{AB} : \cap_{k \in CN_{AB}} V_k &= V_A \cap V_B \cap V_C \cap V_D \\ &= \{A,B,C,D,E\} \cap \{A,B,C,D\} \\ &\quad \cap \{A,B,C\} \cap \{A,B,D,E\} \\ &= \{A,B\}. \end{aligned}$$

If two 1-hop distance nodes i and j satisfy the condition $COND_{ij}$, they can build a secret key. This key is a function of the basic keys provided by nodes in CN_{ij} (i and j's common neighbors set).

Note that if two nodes i and j have their common neighbors set reduced to the minimal set (that is $\{i,j\}$) then they can always establish a secret key by simply composing their basic keys.

Algorithm 1 presents the code executed by node i. It allows the establishment of shared secret keys between node i and its 1-hop neighbors. It is a three steps algorithm. The first communication step (lines 1 and 2) allows each node i to get its 1-hop neighbors (variable V_i). Each node i broadcasts its id that will be received by all the nodes that are within the communication range of node i. Then each node i collects the messages broadcast by its neighbors. In order to simplify the writing of the algorithm, we suppose that each node receives its own messages and handles them in the same way it handles the messages from its neighbors. The second step (lines 3 and 4) allows each node i to discover the neighbors of its neighbors. The local variable $W_i(j)$ of node i contains the set of the neighbors of node j as seen by i.

Note that $W_i(j) = V_j$ (the set of 1-hop neighbors of node j as seen by, respectively, node i and node j). In Line 5, node i computes the set P_i. It is the set of all 1-hop neighbors of i with which it is possible to establish secret keys. Those are the nodes j for which the condition $COND_{ij}$ holds.

In line 6, node i sends a basic key it got by invoking a service get_basic_key(). This could be a random function that returns a basic key of a given fixed size. Each node collects and stores the basic keys it receives in a local variable bk_i, with $bk_i(k)$ being the basic key received by sensor i from sensor k. At this point, node i is able to build the secret keys it shares with its neighbors that are in P_i. For each node $j \in P_i$, node i computes the secret key by combining (a call to comb_keys function) the basic keys sent by all the nodes that are in the common neighborhood to itself and j ($V_i \cap W_i(j)$). The combination of the keys could be a simple xor operation over the basic keys or any other deterministic operation.

On the correctness of the algorithm: The three properties that characterize the SSK problem (Completeness, Symmetry and Privacy) can be stated using the variables of the algorithm.

- Completeness: $\forall i, P_i = V_i \setminus \{i\}$.
- Symmetry: $\forall i, \forall j \in P_i, key_i(j) = key_j(i)$
- Privacy: $\forall i, \forall j \in P_i$, $\nexists k$ s.t. $(V_i \cap V_j) \subset V_k$

It is easy to see that the presented algorithm verifies the two last properties but does not ensure the first one. Indeed, the first property depends on the geographical dispersion of the sensors. The sets of neighbors are a parameter of the problem and the condition $COND$ does not hold for any setting. The simulation section shows that for a uniform dispersion of the sensors, the presented condition $COND$ holds for a large part of the pairs of 1-hop neighbors. Moreover, in the discussion in Section V, we exhibit ad hoc structures for which completeness is ensured (the key establishment is always possible).

Byzantine behavior: If some nodes can exhibit a malicious (Byzantine) behavior as defined in the adversary model (Section II), they cannot interfere much with this key establishment since communication is symmetric. Recall that we assume non cooperating malicious nodes. If a malicious node lies about its neighborhood, since communication is symmetric, a neighborhood disagreement will appear.

In such case, a message of a node j exhibited by a node i is a proof that j and i are neighbors (even if j lies by omitting i). The presence, or the absence of such proof allows to detect the lying node.

A malicious node can announce fictitious neighbors but this will only reduce its chances to share protected links with its neighboring nodes.

B. From One Hop to any Hop

We now have the possibility to establish protected links between neighbors. The objective of this section is to describe how to turn this 1-hop secret keys into a way to securely communicate with (almost) any node in the system.

1) From 1-hop to 2-hop: use disjoint paths: The technique consists in using a disjoint paths classic key establishment technique to create 2-hops shared secrets. Let us first give some definitions. As our network is represented by a graph G, an edge represents a link between the two vertices it connects. An edge is said to be protected if the two nodes it connects share a secret key. A path in G is called *1-hop protected path* if it is composed only of protected edges. Let us consider a path p^1_{ij} that connects node i and node j, then $\mathcal{C}p^1_{ij}$ is the set of the nodes that compose the path p^1_{ij}.

It is possible to establish a 2-hop shared key (a 2-hop protected link) between nodes i and j if there exists two paths that connect i and j such that:

$$COND2_{ij} : \exists \textit{ two 1-hop protected paths } p^1_{ij}, p'^1_{ij} \textit{ s.t.}$$

$$\mathcal{C}p^1_{ij} \cap \mathcal{C}p'^1_{ij} = \{i,j\} \wedge |\mathcal{C}p^1_{ij}| \geq 2 \wedge |\mathcal{C}p'^1_{ij}| = 2$$

If $COND2_{ij}$ is satisfied, node i generates two basic keys bk_i and bk'_i, and sends them over p^1_{ij} and p'^1_{ij}, respectively. These new 2-hop basic keys are sent through the protected paths using the 1-hop secret keys associated to each 1-hop

Algorithm 1 1-hop shared secret keys establishment

(1) send(i) % *the value i is broadcast but will be received only by the neighbors*
(2) Let V_i be the set of received ids % *Set of neighbors including i itself*
(3) send(i, V_i)
(4) Let $W_i(j)$ be the set of ids received from sensor j forall $j \in V_i$ % *neighbors of the neighbors*
(5) Let $P_i = \{j \in V_i \setminus \{i\} | \cap_{k \in V_i \cap W_i(j)} W_i(k) = \{i, j\}\}$
(6) send(get_basic_key())
(7) Let $bk_i(k)$ be the basic key received from sensor k forall $k \in V_i$
(8) Forall $j \in P_i$, $key_i(j) \leftarrow$ comb_keys($\{bk_i(k) | k \in V_i \cap W_i(j)\}$)

link. Thus, only nodes i and j have access to both key parts: each relaying node is able to capture only one of the two parts of the key. Nodes i and j have now a 2-hop protected link to communicate with each other.

2) From 1-hop and 2-hop keys to any-hop: Using previously stated $COND2_{ij}$ condition, it is possible to express the existence of a *2-hop protected path* between nodes i and j in the following way. Let p^2_{ij} be a sequence of nodes starting with i and ending with j. p^2_{ij} is said to be a *2-hop protected path* if any consecutive nodes u and v of the sequence are connected by a 2-hop link, with possible exception of the last pair of nodes that can be connected only by a 1-hop protected path (in order to deal with path length parity).

The idea here may be explained intuitively by the following observation. Let us consider two nodes i and j connected by a 2-hop protected path p^2_{ij}. Consequently i and j are also connected by a 1-hop protected path (p^1_{ij}). Let us consider the path K that can be defined as the complimentary path of p^2_{ij} with respect to p^1_{ij}. This is illustrated on figure 2 where dashed lines represent p^2_{ij} and dotted lines represent K.

More formally and without loss of generality, let us consider $p^1_{ij} =< n_0, \ldots, n_{2k+1} >$ (where $n_0 = i$ and $n_{2k+1} = j$, the size of p^1_{ij} is even). Path p^2_{ij} can be defined as $p^2_{ij} =< n_0, n_2, n_4, \ldots, n_{2k}, n_{2k+1} >$ and path k can be defined as $K =< n_0, n_1, n_3, \ldots, n_{2k-1}, n_{2k+1} >$. Intuitively, we decompose the path p^1_{ij} into two 2-hop disjoint paths that share only the first and the last nodes.

If we consider $\mathcal{K}$ as the set of node that compose the path K then $\mathcal{K} = \mathcal{C}p^1_{ij} - \mathcal{C}p^2_{ij} + \{i, j\}$. In good settings, the 2-hop path K is a 2-hop protected path. Let $\overline{COND2_{ij}}$ be the condition that makes K a protected path. In this case, a p^1_{ij} path defines two 2-hop protected paths that are disjoints.

It is important to notice that the condition the path K has to satisfy ($\overline{COND2_{ij}}$) is not equivalent to $COND2_{ij}$. However, simulations show that, in practice, it is most of time satisfied.

Now suppose i wants to send a message M to j. Let $\{n_1 \ldots n_{2k}\}$ be a set of nodes verifying both $\overline{COND2_{ij}}$ and $COND2_{ij}$. Node i divides M in two parts M_a and M_b such that $M = M_a + M_b$ (whatever the meaning of '+'). Node i sends M_a over path p^2_{ij} (upper part of Figure 2), and M_b over path K (lower part of Figure 2): it ciphers M_a for node n_2 and M_b for node n_1 and sends both parts in one message to n_1. Any node $n_i, (i < 2k-1)$ receiving such a message unciphers half of it (say M_a), ciphers M_a right away for its 2-hop neighbor (n_{i+2}) and relays it to n_{i+1}. The last relaying node, n_{2k}, simply ciphers its part and relays it to j.

Using this technique, nodes i and j may communicate securely through the sensor network without even knowing how far they are *iff* the pair (i, j) satisfies both conditions $\overline{COND2_{ij}}$ and $COND2_{ij}$. Thus, let us define the condition for establishing multi-hop protected paths $COND^+_{ij}$ to be: $COND^+_{ij} = COND2_{ij} \wedge \overline{COND2_{ij}}$

Note that to allow i and j to communicate in a protected way, it is not necessary to know neither the path p^1_{ij} nor the paths p^2_{ij} and K. A greedy 2-hop routing will make it provided that the condition $COND^+_{ij}$ holds.

IV. Simulations Results

The possibility to establish a protected link between two nodes i and j is strongly dependent on node arrangement. The question now is "how often are the required conditions satisfied?". To answer this question, we ran several experiments on a sensor network simulator. The system is modeled by a 500×500 distance units square where 500 sensors are randomly distributed according to a uniform distribution. Various ranges allow us to test different densities (we define the density of the system as the average number of neighbors a node has).

A. Local Point of View

The basic condition this system is based on is $COND$. So an interesting starting point is the average number of times a node i satisfies $COND_i$.

The parameter that will have an effect on the occurrence of satisfied $COND$ conditions is the density of system. Clearly, the more neighbors a node has, the higher the probability to have one or more protected links.

First, one may be interested in the number of protected 1-hop links NL_s we can establish over a system S.

$$NL_s = |\{COND_{ij} | \forall i \in S, \forall j \in V_i\}|$$

Respectively, let NL^2_s be the 2-hop counterpart of this indicator (using $COND2_{ij}$).

Figure 3(a) plots NL_s and NL^2_s as functions of the density of the system. Thus, if the system has an average

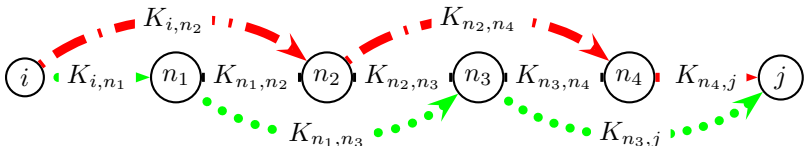


Figure 2. Example of a decomposition of a 1-hop path

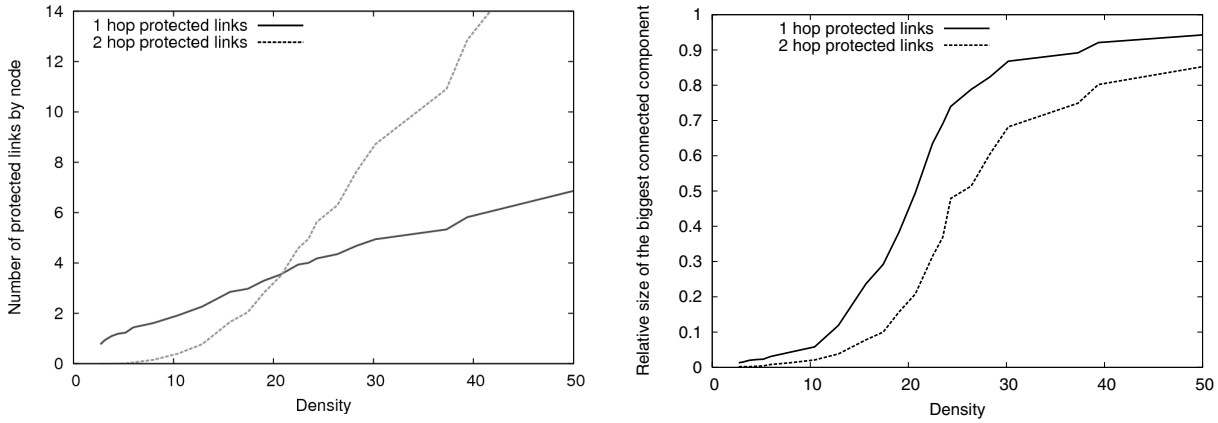


Figure 3. Impact of density on a) the number of protected links of the system b) the connectivity of the protected graph

density of 40 nodes per communication range, a node may expect 5 or 6 one-hop secret keys. This figure shows the impact of density: higher densities reduce the probability for a given pair of neighbors to share a secret key, but by increasing the number of neighbors a node has, it finally increases the number of neighbors a node shares a secret key.

This figure show the slight impact of density on the number of shared keys a node may expect. However, these results give no hint on the protected links connectivity of the system. In other words, is the system reduced to its protected links still connected?

B. Global Point of View

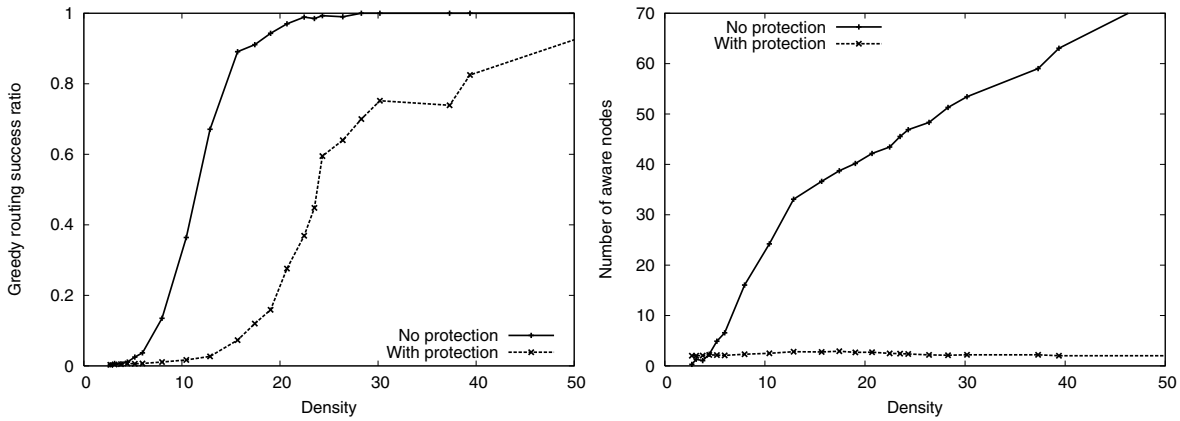
For the communication to be secure, nodes must be able to transmit data across the system using only protected links (using secret keys). An interesting question arises: " What is the ratio of the nodes of the system that can communicate through protected links?". Let $G(V, E)$ be the system graph, as presented in the system model section, such that $(i, j) \in E \Leftrightarrow$ *i and j are 1-hop neighbors.* Let $G_1(V, E_1)$ and $G_2(V, E_2)$ be the two graphs such that $(i, j) \in E_1 \Leftrightarrow COND_{ij}$ and $(i, j) \in E_2 \Leftrightarrow COND2_{ij}$. Let S_1 and S_2 be the largest connected subset of vertices of G_1 and G_2 respectively. Figure 3(b) shows the variations of $|S_1|/|G|$ (dark curve) and $|S_2|/|G|$ (gray curve).

Both curves show a slight percolation effect which illustrates the effect of the density on global connectivity. However, it shows that this algorithm ensures a protected communication within a large part of the network as soon as it is dense enough.

C. Privacy Enhancement

Figure 4 presents interesting results concerning routing using protected links. The experiment is the following: each system node computes its protected links, and tries to send a message to the sink (arbitrarily defined as the closest node to the system center), using either first no protection and then the protected links established by the protocol. Each point of the curve is the average of 20 independent experiments. The left Figure 4(a) plots the success ratio of the greedy routing algorithm over nodes geographical coordinates. It is interesting to observe that greedy routing using protection is efficient as soon as the density is high enough.

The right Figure 4(b) compares the average number of nodes that may listen to a message successfully exchanged between a node and the sink. Using no protection, any node neighboring a node on the message path can listen to it. Using protection, nodes aware of the message are: the node originating the message, the sink node, and eventually nodes between the sending node and a node connected to the sink by a multi-hop protected link. It is interesting to observe that most of the time, the sending node and the sink are directly connected by a multi-hop protected link and can thus confidentially communicate (only two aware nodes). Without protection, the number of nodes aware of an exchanged



(a) (b)

Figure 4. Impact of the protocol over greedy routing and confidentiality

message quickly raises with the density, illustrating the threat for confidentiality of combining radio waves and multi-hop communication in an unsecured network.

V. Discussions

A. Best Case Structures

In the considered model, sensors are disseminated in a uniform but random way. In such setting the probability to establish protected links is high but still lower than 1. It is possible to design structures that allow to establish protected links between any pair of neighbor nodes. In other words, structures whose associated communication graph satisfies property $COND_{ij}$ for all the edges (i, j) of the graph E representing the network. Obviously, we assume that a human intervention is necessary to place each node in a particular place. Figure 5(b) depicts two such regular structures with three different transmission ranges. Recall that a geographical structure alone does not define the communication graph; we need also to consider the transmission range. One can see that for the two proposed structures, the associated communication graphs satisfy the above cited property whatever is the transmission range. The only effect of the transmission range is on the variation of the number of neighbors that participate in the key. In all situations, if we consider any two neighbor nodes (black circles), the intersection of the neighbors of their neighbors only includes the two initial nodes (their common neighbors are presented as rectangles).

B. Regular Structures

We here provide a sufficient condition on the communication graph to ensure confidentiality among all pairs of nodes. Let $G(V, E)$ be a graph, and let V_i be the neighbors of a node $i \in V$. We assume the graph symmetric: $i \in V_j \Leftrightarrow j \in V_i$.

Definition 1: G is locally differentiable iff

$$\forall x \in V, \forall y, z \in V_x^2, (V_x \cap V_y) \neq (V_x \cap V_z)$$

Intuitively, this means that from the point of view of a node, all its neighbors have different neighborhoods. Let $K_{xy} = \bigcap_{k \in V_x \cap V_y} V_k$. Recall that any link $(x, y) \in E$ is protected iff $K_{xy} = \{x, y\}$.

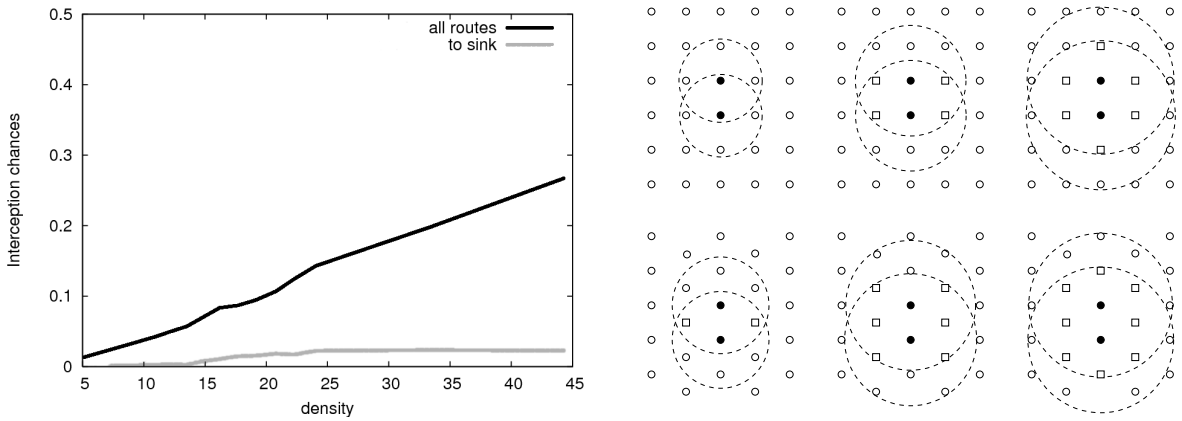
Theorem 1: Let G be locally differentiable. Then for any node $x \in V$, and for any $y \in V_x$, there exists a protected path between x and y p_{xy}^1.

Proof The proof is made by induction on k in the following statement: let x and y be two neighboring nodes in G such that $|K_{xy}| = k$, then there exists a path p_{xy}^1 of protected links between x and y. Let us first observe that $\{x, y\} \in K_{xy}$ since $\{x, y\} \in V_x$ and $\{x, y\} \in V_y$ and that x and y are neighbors. Thus if $|K_{xy}| = 2$, then $K_{xy} = \{x, y\}$: x and y have a protected path of size 1 (a protected link), which proves the statement for $k = 2$.

Assume now that the statement holds for any $h < k$, with $k > 2$. Let x, y be two neighboring nodes such that $|K_{xy}| = k$. Since $k \geq 3$ there exists a node a such that $a \neq x, a \neq y$ and $a \in K_{xy}$.

$$\begin{aligned}
a \in K_{xy} &\Leftrightarrow \forall \ell \in V_x \cap V_y, a \in V_\ell \\
&\Leftrightarrow \forall \ell \in V_x \cap V_y, l \in V_a \\
&\Leftrightarrow V_a \supset V_x \cap V_y \\
&\Rightarrow V_x \cap V_a \supset V_x \cap V_y \textit{ and } V_y \cap V_a \supset V_x \cap V_y \\
&\Rightarrow \cap_{k \in V_x \cap V_a} V_k \subset \cap_{k \in V_x \cap V_y} V_k \\
&\quad \textit{and } \cap_{k \in V_y \cap V_a} V_k \subset \cap_{k \in V_x \cap V_y} V_k \\
&\Rightarrow K_{xa} \subset K_{xy} \textit{ and } K_{ya} \subset K_{xy}
\end{aligned}$$

The graph G is locally differentiable and since $V_x \cap V_a \supset V_x \cap V_y$ *and* $V_y \cap V_a \supset V_x \cap V_y$, there exists a node z



(a) (b)

Figure 5. a) Probability for a mute malicious node to intercept a communication b) Particular structures

such that $z \in V_x \cap V_a, z \notin V_x \cap V_y$ and symmetrically a node z' such that $z' \in V_y \cap V_a, z' \notin V_x \cap V_y$. Note that $z \notin V_y \Leftrightarrow y \notin V_z \Rightarrow y \notin K_{xa}$. Symmetrically for z' we have $x \notin K_{ya}$.

Thus

$$K_{xa} \subsetneq K_{xy} \text{ and } K_{ya} \subsetneq K_{xy}$$

Consequently, $|K_{xa}| < |K_{xy}| = k$ (idem for K_{ya}). By induction on k, we have two protected paths p^1_{xa} and p^1_{ay}. Let $p^1_{xy} = p^1_{xa} + p^1_{ay}$: p^1_{xy} is a protected path between x and y which proves the theorem. $\square_{Theorem}$

Corollary 1: Let G be a locally differentiable connected graph. We have $\forall x, y \in V^2, \exists p^1_{xy}$.

Proof

Since G is connected, any two nodes x and y are connected by a path p_{xy}. Let $< n_1, \ldots, n_k >$ be this path (using $x = n_1$ and $y = n_k$). As G is locally differentiable, $\forall i \in < 1..n-1 >$, there is a protected link $p^1_{i(i+1)}$. Let $p^1_{xy} = \sum_{i=1}^{n-1} p^1_{i(i+1)}$ be the protected path. $\square_{Corollary}$

In the same way and using similar techniques, it is possible to prove that if the graph G is locally differentiable and 2-connected (there are at least two disjoint paths connecting any pair of nodes) then any pair of nodes can be connected by n-hop protected paths.

C. Symmetric Communication

The model given at the beginning of the paper states that communication needs to be symmetric. Indeed, in the perfect case where we want to establish protected links between any pair of nodes this may be mandatory. However, as for a real sensor network this is not always possible, we can relax this assumption. Let us note that a link that does not serve in the establishment of a secret key is not expected to be bi-directional. Moreover, if two links serve in the establishment of a same protected link then having one of them or both of them non bi-directional has the exact same effect: the expected protected link will not be protected.

Let us consider a sensor network with bi-directional communication links. Assume that our algorithm for establishing protected link is able to protect a fraction δ of the total number of communication links. If now we assume the same network but where (1) there is a fraction τ of non bi-directional links among all the links and (2) each non bi-directional link participated to at least one protected link and (3) no two non bi-directional links participate to the establishment of a same protected link. Then instead of having a fraction $delta$ of protected links we will have a proportion of protected links equal to $\delta(1-\tau)$. In the simulations we carried out, it appeared that as several links participate to the same protected links the real proportion of protected links is always greater then $\delta(1-\tau)$. If the proportion of non-bidirectional links is low (less than 10%) the overall performance of our algorithm is maintained.

D. About Byzantine Behavior

In the previously presented model, we address a particular adversary model which could be described as *the participating Byzantine*. In fact, malicious nodes have to execute the algorithm as the other nodes. This assumption is realistic in an application which provides some services. In this case, if malicious nodes want to access to the services they have to simulate good nodes. Poker game is a nice illustration of this principle: spectators are not *in* the game, and thus are not prevented from cheating (*e.g.* by watching all players' hands). In other words, they cannot win anything.

However, an interesting question concerns the *mute Byzantine sensors*, or eavesdroppers. In this enriched model, Byzantine sensors can also stay mute (and thus undetected) during the application, potentially compromising confidentiality by hearing a particular set of keys.

To evaluate the threat, we ran experiments. Then, using the same protocol, we randomly placed Byzantine mute nodes uniformly over the network. For each node, we then computed whether a Byzantine node is able to intercept a confidential multi-hop communication or not. This can be seen as a Monte-Carlo method to evaluate the vulnerable surface of the network. The vulnerable surface represents the area within a Byzantine node can intercept a communication. It is important to notice that to intercept a communication, a Byzantine node has to intercept two consecutive 2-hop protected paths.

Figure 5(a) plots the probability for a mute malicious node to be located in a place where it is able to compromise some data. We considered two cases: the black curve represents the probability for this kind of node to intercept two consecutive links. The gray one represents the probability to intercept two consecutive links taken by a message sent by any node to reach a particular node (called sink in several applications) located in the network center. This shows for example that a randomly located Byzantine, for a 35 nodes average neighborhood density, has 20% chances to intercept two consecutive routes but only 3% to intercept a message to the sink.

VI. Conclusion

We present in this paper a new approach to establish protected communication in a sensor network. This algorithm does not require any initial configuration. Based only on exchanges between neighbors, it is very efficient in terms of the number of messages.

From a first step that establishes 1-hop protected links, we generalized it to provide nodes with multi-hop secret keys with a high probability provided the system is dense enough. The different simulations we carried out show that a 30 nodes neighborhood density, for example, allows 85% of nodes to have protected communication. For multihop transmissions, our algorithm does not requires key establishment exchanges, and can directly be used on top of greedy routing. Moreover, the security guarantees are similar to solutions based on birthday paradox. Indeed, protection is only probabilistic as only a proportion of links are protected however, once a link is protected it is safe. Finally, our algorithm generates few messages, scales, and requires no initial configuration.

VII. Acknowledgments

This work has been supported by ANR project SHAMAN and Euro-NF Network of Excellence.

References

[1] Cachin C., Kursawe K., and Shoup V., Random oracles in Constantinople: practical asynchronous Byzantine agreement using cryptography. *Proc. 19th ACM Symposium on Principles of Distributed Computing (PODC'00)*, ACM Press, pp. 123-132, 2000.

[2] Carman D., Kruus P., and Matt B., Constraints ans approaches for distributed sensor networks security. *Technical report 010*, NAI Labs, The Security Research Division Network associates Inc., September 2000.

[3] Chan H., Perrig A., and Song D., Random key predistribution schemes for sensor networks, *Proc. IEEE Symposium on Security and Privacy (SP'03)*, page 197, Washington, DC, IEEE Computer Society, 2003.

[4] Clark B., Colbourn C., and Johnson D., Unit Disk Graphs. *Discrete Mathematics*, vol 86(1-3):165-177, 1990.

[5] Eschenauer L., and Gligor V., A key-mamangement scheme for distributed sensor networks. Proc. 9th ACM Conference on Computer and Communications Security, pp. 41-47, ACM Press, 2002.

[6] Gilbert S., Guerraoui R., and Newport C., Of malicious motes and suspicious sensors: on the efficiency of malicious interference in wireless networks. *Lecture Notes in Computer Science*, vol 4305:215-229, 2006.

[7] Lamport L., Shostak R.E., and Pease M., The Byzantine generals problem. *Advances in Ultra-Dependable Distributed Systems*, IEEE Computer Society Press, 1995.

[8] O'Dell R., and Wattenhofer R., Theoretical aspects of connectivity-based multi-hop positioning. *Theoretical Computer Science*, vol 344(1):47-68, 2005.

[9] Szczechowiak P., Oliveira L.B., Scott M., Collier M., and Dahab R., Nanoecc: Testing the limits of elliptic curve cryptography in sensor networks. European Conference on Wireless Sensor Networks (EWSN'08), 2008.

[10] Tague P., and Poovendran R., Modeling adaptive node capture attacks in multi-hop wireless networks. *Ad Hoc Networks*, vol 5(6):801-814, 2007.

[11] Yi C., and Agrawal D.P., Improved pairwise key establishment for wireless sensor networks. *Proc. 2nd IEEE Int. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob 2006)*, Montreal, June 19-21, 2006.

[12] Zhu S., Xu S., Setia S., and Jajodia S. Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach. *Proc. 11th IEEE International Conference on Network Protocols (ICNP03)*, IEEE Computer Society, page 326, Washington, DC, 2003.