Journal of Network and Computer Applications 34 (2011) 1380–1397

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Secure and efficient disjoint multipath construction for fault tolerant routing in wireless sensor networks

Y. Challal [a,*], A. Ouadjaout [b], N. Lasla [b], M. Bagaa [b], A. Hadjidj [a]

[a] Université de Technologie de Compiègne, Heudiasyc, UMR CNRS 6599, France
[b] Centre d'Etude et de Recherche sur l'Information Scientifique et Technique, Algeria

ARTICLE INFO

Article history:
Received 1 October 2010
Received in revised form
16 February 2011
Accepted 10 March 2011
Available online 16 March 2011

Keywords:
Wireless sensor networks
Security
Multipath routing

ABSTRACT

In wireless sensor networks, reliability is a design goal of a primary concern. To build a comprehensive reliable system, it is essential to consider node failures and intruder attacks as unavoidable phenomena. In this paper, we present a new intrusion-fault tolerant routing scheme offering a high level of reliability through a secure multipath routing construction. Unlike existing intrusion-fault tolerant solutions, our protocol is based on a distributed and in-network verification scheme, which does not require any referring to the base station. Furthermore, it employs a new multipath selection scheme seeking to enhance the tolerance of the network and conserve the energy of sensors. Extensive analysis and simulations using TinyOS showed that our approach improves many important performance metrics such as: the mean time to failure of the network, detection overhead of some security attacks, energy consumption, and resilience.

1. Introduction

Wireless Sensor Networks (WSN) is a promising technology for gathering real time information in order to monitor a specific area. Their low cost and ease of deployment make them an attractive solution for a plethora of applications in various fields, such as military tracking, fire monitoring, *etc*. They consist of short range sensing devices that collaborate to carry out monitoring measurements to the end users. Sensor nodes are characterized by some intrinsic properties representing important design factors, such as energy constraints, limited computation and storage capacities, *etc*. In addition, many applications require deploying sensors in harsh environments and in large quantities, making very difficult the manual control and the individual monitoring of sensors. Consequently, failures of nodes become *an inevitable phenomenon* which can reduce dramatically the overall network lifetime and make the communication infrastructure unusable.

Some solutions addressing the network lifetime problem are based on *energy-aware routing mechanisms*, which construct paths using some energy metrics (Al-Karaki and Kamal, 2004). The concept behind this family of protocols is to postpone nodes failure as far as possible, but this method is not enough satisfactory since the operation of the whole network is not guaranteed after the occurrence of these failures that are *inevitable*. More elaborate solutions consider node failure as *a normal property of the network* and enhance the network lifetime by providing fault tolerant mechanisms that guarantee normal operation of the network in presence of failures. Major tolerant solutions for WSN and MANET are based on the multipath routing paradigm, which provides each sensor with alternative paths. Different kinds of multipath schemes have been proposed, offering different levels of reliability and fault tolerance (Ganesan et al., 2001; Lou and Kwon, 2006). Among these schemes, building node-disjoint paths has been considered as the most reliable one. Due to the absence of common sensors between node-disjoint paths, a link disconnection will cause at most *a single path to fail* for any sensor in the network. This can contribute greatly in prolonging the network lifetime since failures do not cause a significant impact into the routing view of sensors.

In real deployments, security becomes another important issue (Ouadjaout et al., 2009; Karlof and Wagner, 2003; Djenouri et al., 2005). In presence of malicious nodes, providing sensors with alternative paths is not sufficient to ensure a reliable system. Thus, it is vital to merge intrusion-tolerant solutions with fault tolerant ones in order to obtain a dependable routing layer able to work in any situation.

In the literature, existing *intrusion-fault tolerant* solutions suffer from many problems and shortcomings. Secure protocols trying to find node-disjoint paths consume an important amount of control messages and thus are not adequate to large scale WSN. On the other hand, secure protocols trying to provide a better

* Corresponding author. Tel.: +33 344234429; fax: +33 344234477.
E-mail addresses: ychallal@hds.utc.fr (Y. Challal), aouadjaout@mail.cerist.dz (A. Ouadjaout), nlasla@mail.cerist.dz (N. Lasla), bagaa@mail.cerist.dz (M. Bagaa), hadjidj@utc.fr (A. Hadjidj).

1084-8045/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2011.03.022

scalability suffer from poor level of fault tolerance and do not consider the intersection of built paths leading to nondisjoint routes.

The contribution of this paper is twofold:

- First, we introduce a new approach of multipath routing, called SMRP (*Sub-branch Multipath Routing Protocol*), derived from node-disjoint paths that enhances significantly the network lifetime comparing to the existing solutions. Furthermore, the message exchange between sensors is very optimal since our scheme requires only one message per node to establish a reliable routing topology.
- We have also developed an efficient and lightweight security scheme, named SEIF (*Secure and Efficient Intrusion-Fault tolerant protocol*) based on the above multipath protocol. SEIF differs from existing intrusion-fault tolerant solutions by providing a totally distributed and in-network execution, which does not require referring to the base station for both *route building* and *security checks*.

The remainder of this paper is organized as follows. Most representative solutions addressing the problem of intrusion-fault tolerance are presented in Section 2. Section 3 provides the design goals and a detailed description of the protocol SMRP. Then we analyze the robustness of our sub-branch disjoint multipath construction, with respect to different packet forwarding schemes, in Section 4 In Section 5, we describe our secure and efficient intrusion-fault tolerant solution SEIF. Then we analyze the security properties of SEIF in Section 6. Simulation results are detailed and analyzed in Section 7. Finally, we summarize our work and draw conclusions in Section 9.

2. Related works

There has been a host of research works in multipath routing for sensor networking area in the last few years. Besides improving network resilience, multipath routing is also used for load balancing (Kim et al., 2008) and QoS provisioning (Li et al., 2010). Using multipath routing provides tolerance of node failures along any individual path and increases the network resilience. Node-disjoint multipath routing protocols construct paths with no common nodes/links. This leads to high resilience and fault tolerance since a node failure will threat only one path. However, they usually suffer from control message overhead and a lack of scalability. In Li and Wu (2006), authors proposed a Node-Disjoint Parallel Multipath Routing (DPMR) algorithm. DPMR uses source delay and onehop response mechanisms to construct multiple paths simultaneously. To ensure node-disjointness, only nodes that have not been occupied by other paths forward route requests to their neighbors. In Hou and Shi (2006), authors described LAND, a Localized Algorithm for finding Node-Disjoint paths. LAND constructs a set of minimum cost node-disjoint paths from every node to the Sink. Branch aware routing (Lou and Kwon, 2006) represents an efficient multipath discovery method based on flooding. HSPREAD (Lou and Kwon, 2006) tags route messages with Sink neighbors IDs (roots) and flood these messages to the network. At receiving several requests, a node chose only one branch and forwards it to its respective neighbors. The main drawback of this method is the limited number of discoverable paths. To find more alternative paths, HSPREAD defines a multipath extension flooding phase where nodes from different branches exchange their discovered paths. As a result, HSPREAD discovers more disjoint paths at the cost of more messages exchange. Some researchers aimed to reduce node-disjoint protocols overhead by relaxing the disjointness requirement; they argue that the construction of partially disjoint paths can reduce the energy consumption and control overhead. Ganesan et al. (2001), explored disjoint and braided paths and compared their performances. They showed that braided path protocols overhead is only half the overhead induced by node-disjoint protocols. However, partially disjoint paths are weak since a single node failure causes a broad failure. NC-RMR (Yang et al., 2010) constructs disjoint and braided multipath to increase the network reliability. Furthermore, it uses network coding mechanism to reduce packet redundancy when using multipath delivery. In wireless sensor networks, data is forwarded by nodes and routed to the Sink. Thus, nodes nearer the Sink relay more packets and actively participate in communication. As a result, these nodes expand more energy and are more failure prone due to battery depletion. Considering this fact, some works focused the disjointness only where it has the higher impact. SAR (Sequential Assignment Routing) algorithm (Sohrabi et al., 2000) requires disjointness only in one hop sink neighborhood. To do this, SAR constructs trees departing from each Sink's neighbor by successively branching at each hop. At the end, most nodes will then be part of several trees and have multiple paths disjoint inside the Sink one hop neighborhood. To ensure fault tolerance and failure recovery, SAR implements a localized path restoration mechanism by means of messages exchange between sensors. This leads to an overhead and scalability issues.

Despite existing similarities between intrusion tolerance and fault tolerance design goals, they have traditionally been studied separately (Wang et al., 2006). However, in resource constrained environments such as WSN, combining them in a unique problem can help to reduce the energy consumption of sensors. The first work on an intrusion-fault tolerant approach was the protocol INSENS (Deng et al., 2006). The main idea of this protocol is to enable the sink node to maintain a complete view of the whole communication topology. To achieve this goal, each sensor must send the list of its neighbors to the sink, with *proofs of neighborhood*. These proofs allow the sink node eliminating inexistent communication links that may be injected by malicious nodes. After reception of these proofs, the sink node can build a correct cartography of the current topology. Hence, by using this centralized approach, INSENS can construct the routing table for each sensor. Moreover, the sink has a full control on the routes' quality and can easily build any kind of multipath topology, including node-disjoint paths. Nevertheless, INSENS is not scalable to large networks since it requires a large amount of communication between sensors and the sink. An enhanced version of INSENS (Deng et al., 2006) was proposed to overcome this scalability problem. EINSENS is a totally distributed protocol in which sensors are able to make local decisions to block malicious packets. However, EINSENS builds only one path toward the sink, but the authors *emulated* a multipath routing by deploying several sinks and constructing a single route to each sink. Lee and Choi (2006) proposed SeRINS, a secure multipath protocol consuming lesser messages than INSENS. This enhancement in the communication overhead leaded to attenuation in the level of tolerance offered by its alternative paths, since SeRINS selects routes using the hop count metric only without worrying about their intersection. As described previously, when removing the property of node-disjoint paths, a failure will have a larger impact on the connectivity of the network and the lifetime of the system. SeRINS introduced a novel approach to deal with intruders. Unlike the centralized approach of INSENS, SeRINS employs a trade-off between centralization and total distribution by delegating partial verifications to sensors. Nevertheless, due to this partial information, when a node detects a problem, it cannot make a decision without referring to the sink node. Therefore, the role of the sink node is curative and intervenes only in presence of inconsistent routing information. Chen and Leneutre (2009) defined the security risk as

the percentage of the packets captured by the adversary. They modeled the multipath routing problem as an optimization problem and proposed a polynomial time complexity algorithm to select node-disjoint paths minimizing the worst case security risk. Same efforts have been done to select multipath maximizing the packet delivery ratio under attacks. For solving this problem, authors proposed a heuristic algorithm which uses game theory. Furthermore, they derived a routing solution to achieve a trade-off between route security and delivery ratio in worst scenarios. However, the proposed solution remains theoretical and could not be easily adapted to the resource constrained wireless sensor networks. Moreover, the solution assumes that each node has a complete knowledge of network topology which is a too strong assumption in the case of WSN since it requires too much exchanges to build this global view of the topology. Nasser and Chen (2007) proposed SEEM (Secure and Energy-Efficient Multipath routing protocol) which finds both braided and disjoint paths. In SEEM, the network adopts a Client/Server scheme, where the Sink (server) does the paths discovery, paths selection and paths maintenance in a centralized way. Hence, the Sink should have the whole network topology. This requires that each node unicasts its neighbors list to the Sink, which consumes much energy and induce huge overhead.

A thorough analysis and overview of secure multipath routing approaches for wireless sensor networks can be found in Stavrou and Pitsillides (2010). Through analyzing existing solutions, one can conclude that intrusion-fault tolerant approaches do not provide an acceptable trade-off between the level of fault tolerance and the induced communication overhead.

In what follows we present our approach consisting of a novel multipath routing construction under the assumption of a security perimeter, that enhances significantly the network lifetime comparing to the existing solutions, without inducing extra message exchange overhead. The secure version of our protocol differs from the existing intrusion-fault tolerant solutions by providing a totally distributed and in-network execution, which does not require referring to the base station for both route construction and security checks.

3. Sub-branch Multipath Routing Protocol

In this section, we describe our protocol *Sub-branch Multipath Routing Protocol (SMRP), a sub-branch disjoint* path construction for one-to-many communications paradigm. In Section 5, SMRP will be employed as the route construction scheme for our intrusion-fault tolerant protocol SEIF.

3.1. Problem definition

Redundancy represents an important concept in the design of a reliable and fault tolerant system. For that reason, node-disjointness have been the most preferable metric in existing multipath routing protocols. There exist different solutions for finding node-disjoint paths between communicating nodes (Lou and Kwon, 2006; Deng et al., 2006; Xiuli and Haibin, 2006). *Branch-aware route discovery* represents an efficient method that fits well the properties of the many-to-one communication paradigm of WSN. This method can be incorporated into the simplest flooding-based protocol, like the TinyOS beaconing protocol, without any additional message requiring only one transmission per sensor (Lou and Kwon, 2006). The main idea of this type of routing is based on *tagging* any route message with the identification of the sink's neighbor that relayed the message. These neighbors are named *root nodes*, and the sub-tree of each one of them is named a *branch*. Using these tags, any sensor can easily decide if two paths are disjoint by comparing the identifier of the root nodes in each path (see Fig. 1).

However, the main drawback of this method is the limited number of *discoverable* alternative paths. Indeed, the ability of discovering new paths by the branch-aware flooding is limited to nodes that have *cousin neighbors*, *i.e.* two neighbors belonging to two distinct branches. To deal with this limitation, H-SPREAD (Lou and Kwon, 2006) proposed an extension to find more extra routes at cost of additional messages, by breaking the property of using "one message per node". When a sensor node discovers a new alternative path, it informs its neighborhood about it. Recursively, this information is propagated through the network to maximize the number of disjoint paths per node. Naturally, this extension overburdens sensors with considerable energy consumption due to the exchange of the extra messages.

3.2. Overview of our solution

In our solution, we have chosen to preserve the constraint of using "one message per sensor". To enable more alternative paths, we have *carefully* redefined the nature of the alternative paths without altering their level of tolerance.

In existing solutions, sensors reject automatically any message from an already discovered branch, in order to maintain the paths node-disjoint. Therefore, a sensor can accept only one route per branch. To explore more routes without adding new messages, we

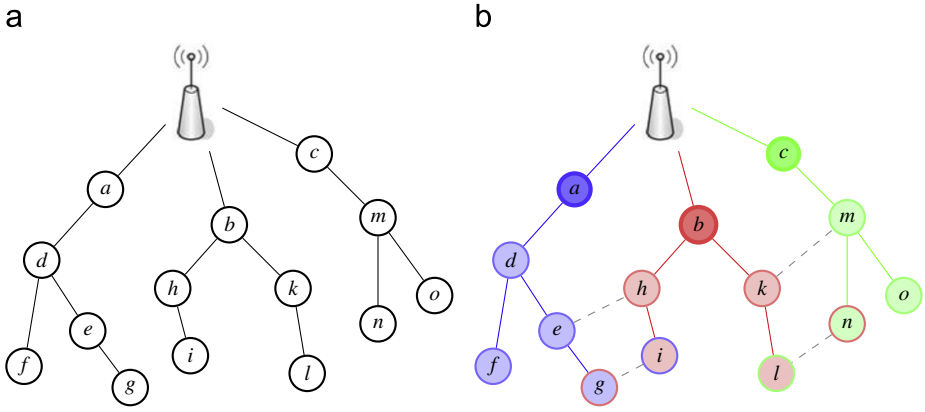

Fig. 1. The concept of branch-aware flooding. (a) A topology obtained by a simple flooding protocol like the TinyOS beaconing protocol; (b) In a branch-aware protocol, the redundant reception of construction message can be exploited to discover new paths, without adding new messages. For instance, when node *g* broadcasts its message, the node *i* can discover an alternative path via the blue branch, since *i* already belongs to the red one. Nodes *g* and *i* are said to be *cousin neighbors*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

have alleviated this constraint by allowing some particular nodes as intersection between paths.

The basic idea of our approach called *Sub-branch Multipath Routing Protocol* (SMRP) follows from the following fact. Root nodes (sink's neighbors) represent the comparison factor between routes in node-disjoint protocols, since two routes are said of the same quality if they came from the same root node. Since the number of root nodes is constant during a round, discoverable alternative paths is limited by the cardinality of this set of nodes. Instead of tagging routes with the roots' IDs, we have chosen to assign the tagging responsibility to the neighbors of root nodes, *i.e.*2-hops neighbors of the sink node. This way, we will construct more alternative routes by allowing root nodes as intersection between routes without adding extra messages. Neighboring nodes of roots can become *sub-roots* and thereby construct their own *sub-branches* (see Fig. 2). A sensor will accept paths within the same branch only if they come from different sub-branches. Therefore, we will not *blindly* reject routes within the same branch in order to avoid intersection at roots level. In fact, allowing such *controlled intersection* will increase the tolerance of the system and improve the survivability of the system. Indeed, our simulations showed that the amelioration of the MTTF offered by SMRP, comparing to the results of H-SPREAD, ranges from 6% to 44% depending on the nature of deployment.

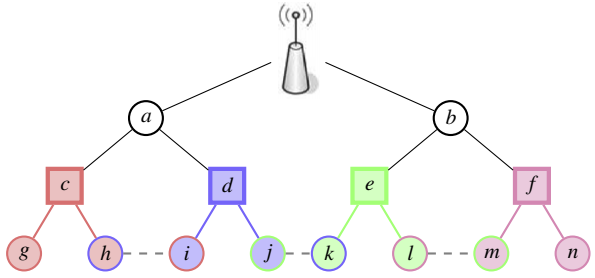

Fig. 2. Sub-branch construction of the protocol SMRP. Routes are tagged with the IDs of 2-hops neighbors of the sink. Theses nodes are named sub-roots and represented by the square nodes. When two nodes advertise two distinct sub-branches, they become *cousin*. In this example, we can distinguish between two types of cousin neighbors. The nodes *j* and *k* belongs to two distinct branches, hence they can share totally disjoint paths. However, the nodes *h* and *i* belongs to the same branch but to two distinct sub-branches. In this case, they can share two routes having the same root node in common.

3.3. Motivation

Tolerating paths intersection at root nodes is not a strong assumption. This can be motivated by the fact that base station neighbor nodes remain accessible for maintenance. In some kind of deployments they may be even equipped with permanent energy supply, or can be powerful nodes whenever affordable given their reduced number. As illustrated in Fig. 3, it is common to assume a security perimeter with a radius *r* around the base station.

In our case, we assume that the radius of the security perimeter covers only one hop neighbors of the base station, which is a reasonable assumption for most common applications of WSN.

3.4. Terminology and notation

In Table 1 we introduce the different notations and terminology used in the description of our protocols operation.

Table 1
Notations.

Notation	Description
F	A one way function
$E(K,m)$	Encryption of m using key K
$A \rightarrow B : m$	A sends to B the message m
$A \rightarrow * : m$	A broadcasts the message m
$K_{A,B}$	Secret pair wise key between A and B
BK_A	Broadcast key of node A
$\|$	Concatenation
OHC	One way Hash Chain

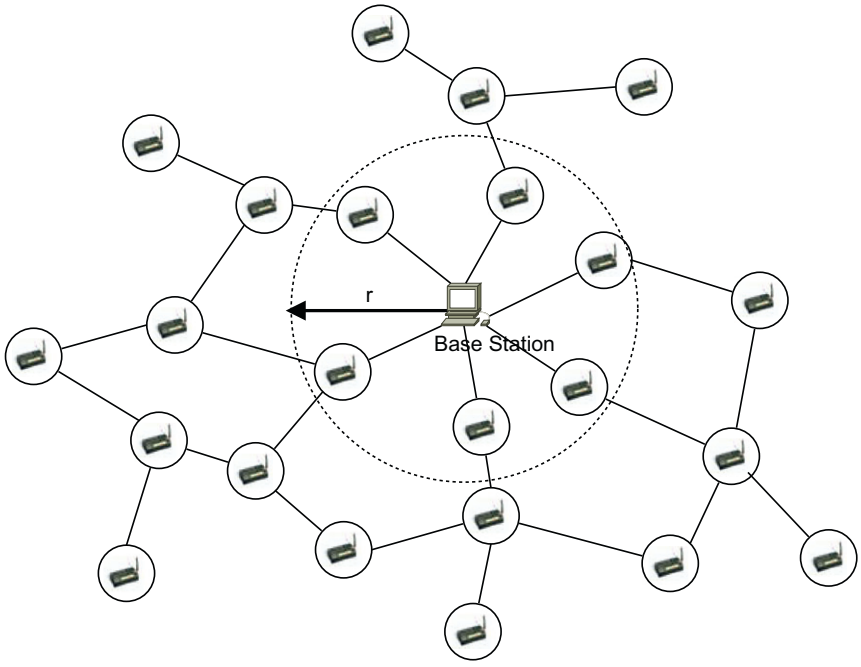

Fig. 3. Security perimeter with radius *r* around the base station.

3.5. Description

We assume that the sink node broadcasts periodically a *Route REQuest (RREQ)* message to discover paths relating each sensor node to the sink. The RREQ (*Route REQuest*) message has the following format:

(*r*,*parent*,*subBranch*)

where:

- *r*: the sequence number identifying the current round.
- *parent*: the ID of the sending node.
- *sub-Branch*: the ID of the sub-root, *i.e.* the second sensor having relayed this RREQ.

Each sensor maintains a routing table containing an entry for each fresh alternative path. Each entry indicates the ID of the parent and the ID of its sub-branch.

3.5.1. Round initialization

Periodically, the sink starts the construction of a new tree by broadcasting the following message:

$sink \longrightarrow * : r, sink, \varnothing$

3.5.2. Selection of alternative routes

When a sensor receives a message indicating a new round, it initializes its routing table by removing any discovered path. The sensor also starts a random *decision timer* that defines the discovery period of alternative paths before relaying the RREQ message.

Upon receiving sub sequent RREQ messages in the same round, the sensor should verify their intersection with already discovered paths. If the received sub-branch tag does not exist in the routing table, the sending node is selected as an alternative parent and the new route is added to the routing table. Otherwise, the message is ignored since it does not fulfill the required quality.

3.5.3. Routing decision

During each round, every sensor should relay the RREQ message only once. When the decision timer fires, the sensor must choose its main parent among the discovered alternative paths and relays this decision to its neighborhood. This choice is done in three levels:

- If the sensor received a RREQ from the sink node during the current round, the sensor becomes a new root node and sends the following message:

  $i \rightarrow * : r, i, \varnothing$

- Otherwise, the sensor searches its routing table to check whether it has received a RREQ with an empty sub-branch. If such entry exists, the node becomes a sub-root and broadcasts the following message:

  $i \rightarrow * : r, i, i$

- Otherwise, the node selects randomly an entry from its routing table and sends the following message:

  $i \rightarrow * : r, i, sbId$

  where *sbId* represents the ID of the sub-branch of the selected entry.

Following these steps, *sub-branch disjoint* paths relating each sensor node to the sink are discovered (cf. Fig. 2). Under the assumption that neighbor nodes of the sink are somehow more reliable than other nodes of the network, as explained above, this approach allows discovering more disjoint paths and consequently enhances the overall availability of the network. Nevertheless, SMRP is a basic version of our sub-branch disjoint paths construction approach. It may be used only in safe environments, since it does not provide some security properties such as: round initialization authentication, sub-branch tags authentication, parent authentication, and freshness. In Section 5, we present a secure version of SMRP that we call *Secure and Efficient Intrusion-Fault tolerant routing protocol (SEIF)*.

Before that, we analyze in what follows the robustness of our sub-branch aware multipath construction approach coupled with different data forwarding techniques; namely: full duplication of packets, random selection of the parent forwarding data packets, and the partial (k,n)-loss tolerant duplication of packets.

4. Analysis of sub-branch aware multipath routing

In this section we will give an insight on the reliability of various routing approaches used in WSN. In particular, we will consider different multipath constructions combined with different ways to use the constructed paths. This allows to show the best marriage between the underlying topology and the data dissemination method with respect to successfully sending a packet to the sink node.

We will consider the following multipath constructions:

- Node-disjoint multipath construction
- Sub-branch disjoint multipath construction

In what relates to using the constructed multipath routes, we will consider the following techniques:

- *Full duplication*: in this technique a packet is first duplicated and then sent through all the constructed paths to the sink.
- *Random parent selection*: in this technique one parent is selected among the different parents belonging to the different paths, and used to forward the packet to the sink node.
- *(t,n)-loss tolerant duplication*: in this scheme, a packet is processed using a specific (t,n)-loss tolerant algorithm such as IDA (Rabin, 1989). A (t,n)-loss tolerant algorithm provides as a result n pieces of information such as only $t < n$ pieces are required to reconstruct the original packet. All the n pieces are sent through the different constructed paths. If the sink receives at least $t < n$ pieces, it would reconstruct the original packet using the (t,n)-loss tolerant algorithm.

4.1. Notations and assumptions

We calculate the probability f_i that the source node i becomes disconnected from the sink with respect to the used combination of multipath construction technique and data forwarding technique. A path between i and the sink is considered disconnected if at least one node on the path is failed. Node i is considered disconnected from the sink, if i has no mean to send packets to the sink given the used data forwarding technique.

Considering that the objective of this section is to provide an insight on reliability of the different schemes and not an exhaustive study, and for simplicity reasons we consider only independent node failures. In this model, each node has a probability of failure α during a small interval T. Isolated failures are not

completely divorced from reality. They can represent failure due to energy dissipation or localized environmental effects at low deployment densities (Ganesan et al., 2001).

We will use the following notations to calculate f_i for each case:

- α: probability of independent node failure.
- α_r: the probability of independent failure of root nodes. We assume that $\alpha_r < \alpha$. Indeed, using sub-branch multipath construction is legitimate only in the case we assume there is a mean to make root nodes more reliable than the other nodes of the network. This is possible whenever those nodes belong to a security perimeter where nodes are easily accessible for maintenance, or are more powerful nodes with more energy, etc.
- π_i: a path relating node i to the sink.
- $|\pi_i|$: number of hops in the path π_i.
- $\{\pi_i\}$: set of paths relating node i to the sink.
- $\{\pi_i(n)\}$: set of paths relating i to the sink passing through node n.
- $roots_i$: set of root nodes from sink down to i.
- $subs(\{\pi_i\},k)$: all sub-sets of size k from the set of paths $\{\pi_i\}$.
- $F(\{\pi\})$: the event that the set of paths $\{\pi\}$ have failed. A path is considered failed if at least one node on the path is failed.

4.2. Node-disjoint paths construction

Assume that multiple node-disjoint paths are constructed between a sensor i and the sink. In what follows, we calculate f_i for the data relay techniques cited above:

4.2.1. Full duplication

Having a set of alternative parents from distinct branches, a sensor can send the same data over all its routes to ensure the maximum reliability.

Proposition 1. *Consider a node-disjoint multipath construction from a sensor node i toward the sink. If the node i uses full duplication to send a packet toward the sink, the probability that node i becomes disconnected from the sink is equal to*:

$$f_i = \prod_{\pi \in \{\pi_i\}} (1-(1-\alpha)^{|\pi|}) \tag{1}$$

Proof. Cf. Appendix A.

4.2.2. Random parent selection

In order to load balance transmission energy over the nodes of the network, a sensor node may choose randomly one of its parents to transmit a packet to the sink.

Proposition 2. *Consider a node-disjoint multipath construction from a sensor node i toward the sink. If the node i uses a one randomly chosen parent to send a packet toward the sink, the probability that node i becomes disconnected from the sink is equal to*

$$f_i = \sum_{\pi \in \{\pi_i\}} (1-(1-\alpha)^{|\pi|}) \times \frac{1}{|\{\pi_i\}|} \tag{2}$$

Proof. Cf. Appendix A

4.2.3. (t,n)-loss tolerant duplication

This scheme tolerates the disconnection of at most n-t paths among the n paths used to send the n pieces resulting from applying the (t,n)-loss tolerant algorithm to a packet.

Proposition 3. *Consider a node-disjoint multipath construction from a sensor node i toward the sink. If the node i uses (t,n)-loss tolerant algorithm to process a packet and then sends the resulting n pieces toward the sink over n disjoint paths, the probability that node i becomes disconnected from the sink is equal to*

$$f_i = \sum_{k=n-t+1}^{n} \sum_{s \in subs(\{\pi_i\},k)} \prod_{\pi \in s} (1-(1-\alpha)^{|\pi|}) \times \prod_{\pi \in \pi_i - s} (1-\alpha)^{|\pi|} \tag{3}$$

Proof. Cf. Appendix A

4.3. Sub-branch disjoint paths construction

In this scheme we tolerate that root nodes (nodes at one hop from the sink) become points of intersection between disjoint sub-branches. We have illustrated in the previous section (SMRP) how to construct such disjoint sub-branches using tags of sub-root nodes.

4.3.1. Full duplication

In this scheme a packet is duplicated then sent through the different disjoint sub-branches. The disconnection of source node i from the sink happens when all disjoint sub-branches become disconnected from the sink. Failure of sub-branches belonging to the same branch is dependent since the sub-branches would have the root node in common.

Proposition 4. *Consider a sub-branch disjoint multipath construction from a sensor node i toward the sink. If the node i uses full duplication to send a packet toward the sink, the probability that node i becomes disconnected from the sink is equal to*

$$f_i = \prod_{r \in roots_i} \left(\alpha_r + (1-\alpha_r) \times \prod_{\pi \in \{\pi_i(r)\}} (1-(1-\alpha)^{|\pi|-1}) \right) \tag{4}$$

Proof. Cf. Appendix A

4.3.2. Random parent selection

In this case, one parent is selected randomly to forward the packet to the sink.

Proposition 5. *Consider a sub-branch disjoint multipath construction from a sensor node i toward the sink. If the node i uses a one randomly chosen parent to send a packet toward the sink, the probability that node i becomes disconnected from the sink is equal to*

$$f_i = \sum_{\pi \in \{\pi_i\}} (1-\alpha_r)(1-(1-\alpha)^{|\pi|-1}) \times \frac{1}{|\{\pi_i\}|} \tag{5}$$

Proof. Cf. Appendix A

4.3.3. (t,n)-loss tolerant duplication

Recall that in this case a packet is processed and the resulting pieces are sent over n disjoint sub-branches. Only $t < n$ pieces are required to reconstruct the original packet.

Proposition 6. *Consider a sub-branch disjoint multipath construction from a sensor node i toward the sink. If the node i uses (t,n)-loss tolerant algorithm to process a packet and then sends the resulting n pieces toward the sink over n disjoint paths, the probability that node i becomes disconnected from the sink is equal to*

$$f_i = \sum_{k=n-t+1}^{n} \left(\sum_{s \in subs(\{\pi_i\},k)} \prod_{r \in roots_i} G_i(s,r) \right) \tag{6}$$

where

$$G_i(s,r)=\begin{cases}\alpha_r+(1-\alpha_r)\times\prod\limits_{\pi\in\{\pi_i(r)\}}(1-(1-\alpha_r)^{|\pi|-1}),\\ \textit{if } \pi_i(r)-s=\emptyset\\ (1-\alpha_r)\times\prod\limits_{\pi\in\{\pi_i(r)\}\cap s}(1-(1-\alpha_r)^{|\pi|-1})\times\\ \prod\limits_{\pi\in\{\pi_i(r)\}-s}(1-\alpha_r)^{|\pi|-1},\ \textit{if } \pi_i(r)-s\neq\emptyset\end{cases}$$

Proof. Cf. Appendix A.

4.4. Analysis and comparison

We have simulated, using TOSSIM/TinyOS (Philip et al., 2003; Hill et al., 1979), a node-disjoint routing protocol (HSPREAD, Lou and Kwon, 2006), and our sub-branch disjoint routing protocol (SMRP). Then, using the formulas presented above, we have calculated the probability of node disconnection from the sink with respect to the three data forwarding schemes. Finally, we calculated the average probability of node disconnection from the sink with respect to the number of hops between each node and the sink. We have considered the following settings:

- Network size: 200 sensor nodes randomly deployed over a surface.
- Average network density: 20.
- $\alpha = 0.2$.
- $\alpha_r = 0.02$.

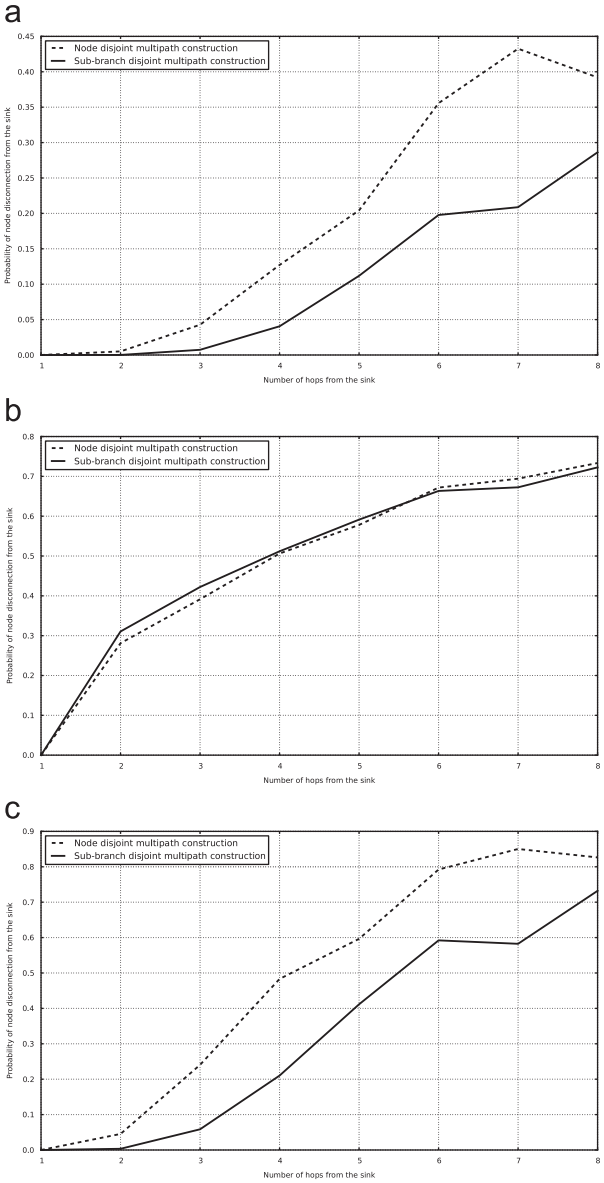

Fig. 4. Node disconnection probability for node vs. sub-branch disjoint multipath constructions. (a) Full duplication, (b) random selection of parent, (c) (2, *n*)-loss tolerant duplication.

Figure 4 illustrates the evolution of disconnection probability depending on the number of hops separating a node from the sink.

As predicted, we remark that the disconnection probability is smaller in the case of sub-branch disjoint multipath construction using full duplication and (2,*n*)-loss tolerant schemes. This is due to the fact that the number of sub-branch disjoint paths is greater than the one of node-disjoint paths. Of course, sub-branch disjoint multipath construction is justified only with the assumption that we have some security perimeter around the base station where making onehop roots accessible for maintenance is affordable .

5. Secure and Efficient Intrusion Fault tolerant routing protocol

The *Secure and Efficient Intrusion Fault tolerant routing* protocol (SEIF) is a merge between the multipath topology built using SMRP and an efficient in-network sub-branch authentication mechanism that we propose in what follows. This merge brings up a highly reliable and secure routing system tolerant to failures and attacks.

5.1. Problem definition

Despite the numerous advantages of branch-aware flooding mechanisms, this efficient concept is prone to different types of attacks due principally to the unauthenticated branch tagging. For instance, an intruder can advertise some messages tagged with inexistent branches in order to attract the maximum number of paths and become an important router among relaying sensors. This predominant position gives the intruder control over a considerable amount of traffic flow, which is very dangerous in many applications.

To defend against these attacks, it is necessary to provide security services that allow verifying the authenticity and freshness of claimed sub-branches. In summary, this mechanism should verify the following requirements:

- *Sub-branch origin authentication*: Sensors should be able to verify if the claimed sub-branches are really rooted at trusted sub-roots. This authentication should be one-to-many requiring some asymmetric properties. In other words, any sub-root should provide a proof that other sensors can only verify, without being able to generate it in advance.
- *Freshness*: To protect against replay attacks, sensors must verify the freshness of exchanged messages.
- *Deployment-independence*: This is an important property because the sub-roots are only known after deployment. Moreover, the number of these sub-roots can vary over time while removing or adding sensors.
- *Energy conservation*: The security verifications should employ lightweight computations avoiding the use of public key cryptography or an excessive communication.

Unfortunately, sub-branch tags are not the only vulnerable information to protect. Any tree-based routing protocol must

provide two principal mechanisms: verification of round initialization and parent authentication.

For long time deployments, tree reconstruction is *unavoidable*, even with a fault tolerant solution since sensors can be added to the network. An attacker may exploit this property by sending a forged round initialization to spoof the sink's identity. As a result, new paths are created towards the intruder, giving him a total control over sensed data. Therefore, it is important to ensure that the sink is the unique starting point of any tree construction attempt.

The second information to protect is the parent ID. Since a WSN may contain powerful intruders, an attacker may use a high-powered transmitter to reach a large set of nodes, to make them believe that they are neighbors of him while they are not. To defend against this Hello Flooding attack, each sensor should discover its reachable neighborhood, consisting of neighbors having a bidirectional link, using a challenge-response mechanism (Deng et al., 2006; Karlof and Wagner, 2003).

5.2. Protocol overview

SEIF is a secure version of SMRP. Indeed, SMRP does not provide round initialization authentication, sub-branch tags authentication, parent authentication, and freshness.

SEIF provides answers to these authentication requirements through an efficient symmetric cryptography mechanism called: One way Hash Chain (OHC) (Lamport, 1979). Particularly, round sequence numbers, sub-branch tags, and parent requests, are authenticated using one way hash chains.

Definition 7. A one way hash chain is a sequence of numbers $(K_i)_{0 \le i \le n}$ generated by a one way function F as follows:

$\forall i, 0 \le i < n : K_i = F(K_{i+1})$

where K_n is a random number generated by the sink. The security of this concept is based on the fact that knowing K_i, it is computationally infeasible to determine K_{i+1}.

What makes one way hash chains interesting in our case, is its ability to provide broadcast authentication (authentication of messages coming from the sink) without requiring to share a different key between the sink and each sensor in the network.

Indeed, before network installation, a set of hash chains are generated and stored in the sink. During the execution of the protocol, each sensor maintains a *chain verifier* for every OHC. For a node i, a chain verifier $CV_{i,j}$ represents the last known value of the jth chain. This variable is initialized with the first unused value of the corresponding chain, and uploaded into sensors before deployment. Each OHC can be considered as a *generator of one-way sequence numbers*. Each sequence number will allow authenticating a specific information: new round initialization, sub-branch tag distribution, a parent route request forwarding, as explained in the following section.

We assume that the sink shares a symmetric key with each root node, and each root shares a symmetric key with each sub-root. These keys will provide confidentiality while distributing the sub-branch tags to the sub-roots. As we will see below, this twohops only encryption is required to avoid that an intruder replays a sub-branch tag into another sub-branch and hence reduces the number of constructed sub-branch disjoint routes to the sink. We assume also that each node i shares one local broadcast key BK_i with its onehop neighbors. This key will be used to distribute to i's neighborhood the first verifier $NV_{(*,i)}$ of the one way chain used by i to provide authenticity for its subsequent requests. Indeed, this secure distribution of the first value of this one way chain cannot be done before deployment, and hence we require a cryptographic protection through encryption for its distribution. The above symmetric keys can be established using some existing key management schemes (Xiao et al., 2007).

5.3. Detailed description

5.3.1. Bootstrapping

The main purpose of this phase is to initialize the different types of chain verifiers. Every sensor i maintains three types of verifiers:

- A special round verifier RV_i is reserved to authenticate round initializations.
- For sub-branch authentication, node i maintains for each chain j a branch verifier $CV_{i,j}$ and the position $P_{i,j}$ of that value within its corresponding chain. Note that the round and sub-branch OHCs are stocked in the sink node. When a sensor is deployed in the network, it is pre-loaded with the first unused value of each chain.
- For each reachable neighbor j, node i maintains a *neighbor verifier* $NV_{i,j}$. When a sensor is deployed, the administrator pre-loads it with its local chain for one hop authentication. After establishment of the broadcast key BK_i, node i reveals its first unused value V:

$$i \rightarrow * : i, E(BK_i, V) \tag{7}$$

As described previously, the encryption of V with BK_i enables a reachable neighbor j to initialize its verifier $NV_{j,i}$. If node j is not a newly deployed sensor and i represents a new neighbor to j, the latter should reply with the last used value of its local chain. Since many sensors may be deployed together, j should wait for random period of time before sending its value to inform all newly deployed sensors with a unique message.

5.3.2. Tag distribution

The goal of this phase is to provide each sub-root with its valid tag. Since sub-roots are two hops away from the sink, the latter should select a set of relay nodes among root nodes to transfer these tags. This can be achieved by constructing a dominating set DS from the set of root nodes covering the 2-hops neighborhood. After the construction of DS, the sink will send to each node $i \in DS$ a ring of values from distinct chains:

$$\begin{aligned} sink \rightarrow i : \quad & E(K_{sink,i}, subRoot_1 \| n_1 \| p_1 \| V_1 \\ & \| \ldots \| \\ & subRoot_m \| n_m \| p_m \| V_m \| R) \end{aligned} \tag{8}$$

where:

- $subRoot_k$ represents the ID of one sub-root covered by node i.
- n_k is the ID of the chain affected to $subRoot_k$ during the current round[1].
- V_k is the first unused value of the chain n_k.
- p_k is the position of V_k within the chain.
- m is the number of sub-roots covered by node i.
- R represents the round sequence number.

When a root node i receives the message (8), it must verify if $RV_i = F(R)$. In case of incorrect round sequence number, the message is ignored. Otherwise, the round verifier RV_i is updated. Then, node i authenticates the received branch tags. For each tag

[1] Because SEIF is independent from deployment, the chains are not intrinsically linked to sub-root nodes. From a round to another, the affectation of chains to these nodes can change without disturbing the execution of the protocol.

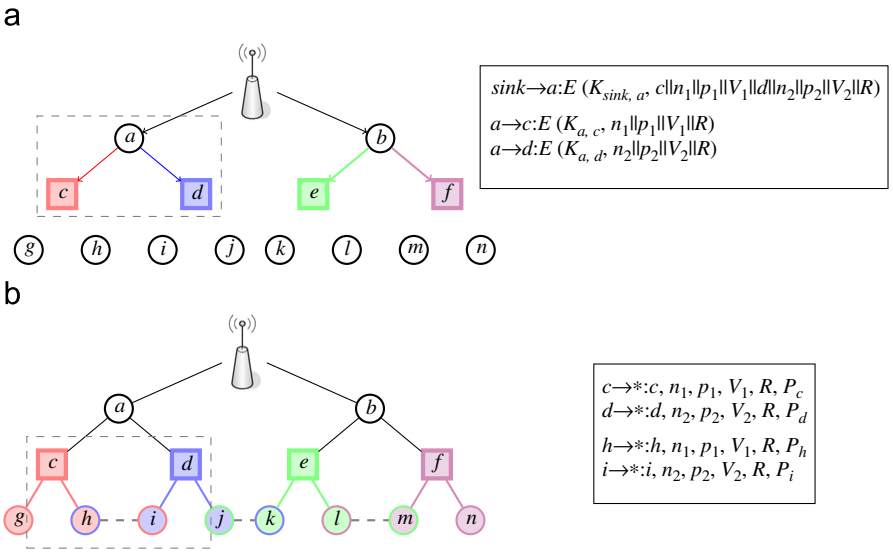

Fig. 5. An example of the secure sub-branch aware flooding provided by SEIF. (a) This figure describes the tag distribution inside one branch, delimited with the dashed square. The same process is executed in the second branch. The sink begins by sending a ring of valid tags (V_1 and V_2) to root node *a*. The latter authenticates the sink node and verifies the round and sub-branches sequence numbers. After that, each sub-root node, represented with a square, decrypts its own tag to start the construction of its sub-branch. (b) At the beginning of the *tree construction* phase, sub-root nodes *c* and *d* broadcast their respective tags, which will be propagated through the sensor network to establish a secure multi-path topology. At each reception of a control message, sensors perform three types of verification: *parent authentication*, *round verification* and *sub-branch authentication*. For instance, when sensor *h* receives the message broadcast by *c*, *h* verifies the following equations: (1) $NV_{h,c}=F(P_c)$, (2) $RV_h=F(R)$ or $R=RV_h$, (3) $CV_{h,n_i}=F^{p_1-P_{h,n_1}}(V_1)$. When two neighbors advertise two distinct valid tags, they become *cousins*.

V_k, i should verify two conditions:

$$\begin{cases} p_k > P_{i,n_k} \\ CV_{i,n_k} = F^{p_k - P_{i,n_k}}(V_k) \end{cases}$$

The variables P_{i,n_k} and CV_{i,n_k} are updated accordingly. The final step during tag distribution is the relay of each tag to the target sub-root using the following message:

$$i \rightarrow subRoot_k : E(K_{i,subRoot_k}, n_k \| p_k \| V_k \| R) \quad (9)$$

After a sensor decrypts the message (9) and verifies the round and sub-branch sequence numbers (using the same procedures as described above), it can start the creation of its own sub-branch. Using the provided tag, it can now pretend to be a sub-root for the current round. An example describing the different steps during the tag distribution phase is presented in Fig. 5(a).

5.3.3. Tree construction

Sub-roots start the construction of their sub-trees by advertising the following message:

$$i \rightarrow * : i,n,p,V,R,P_i \quad (10)$$

where :

- n, p, V and R represent the values received from the root node within the message (9).
- P_i is the first unused value of the local OHC for one hop authentication.

When a sensor j receives the message (10), it authenticates the sending node by verifying if P_i represents the next sequence number of the neighbor verifier $NV_{j,i}$, *i.e.* $NV_{j,i}=F(P_i)$. After successful authentication and update of $NV_{j,i}$, node j verifies the round sequence number R. If $RV_j=F(R)$, the sensor node updates its round verifier and reinitializes its routing table by removing all its alternative paths. Contrary to messages (8) and (9), node j also accepts the received message if $R=RV_j$ (*i.e.* message belonging to the current round) in order to discover alternative paths.

The next step is to authenticate the sub-branch tag. If the received tag verifies the following two conditions:

$$\begin{cases} p > P_{j,n} \\ CV_{j,n} = F^{p-P_{j,n}}(V) \end{cases}$$

the sensor elects the sending node as an alternative parent and updates $CV_{j,n}$ and $P_{j,n}$ with the received values. However, some malicious nodes can exploit the repetitive execution of the function F to launch an energy exhaustion attack. An intruder can advertise a message with a correct round sequence number but with a large value of p in order to force neighboring nodes to carry out a lot of hash calculations. To avoid this form of denial of service attacks, we have added the following condition:

$$p - P_{j,n} < D$$

where D defines the maximum number of iterations over function F to verify whether the received value belongs to the claimed chain.

As described for the protocol SMRP, sensor j launches a random timer to relay its routing decision when it detects a new round. When this decision timer fires, the sensor node chooses randomly one main parent among the discovered alternative paths, and sends the message (10) using the sub-branch tag of the chosen main parent. Figure 5(b) gives an example of tree construction and alternative path discovery in the protocol SEIF.

6. Security properties analysis

In this section, we present analysis of the security properties provided by SEIF protocol

Proposition 8 (*Authentication of round initialization*). *SEIF guarantees authentication of round initialization. In other words, an intruder cannot initialize the construction of a new tree on behalf of the sink.*

Proof. The sink uses a one way hash chain to generate round sequence numbers. Each round is identified with a sequence number that is the last delivered value of the one way hash chain. Hence, nodes of the network can verify the new round

sequence number R_{i+1} through checking whether $R_i = F(R_{i+1})$, where R_i is the previous round sequence number. However, they cannot calculate the following sequence number.

Indeed, an intruder cannot use the current round sequence number to calculate the following sequence numbers since it is computationally infeasible to calculate R_{i+1} knowing R_j with $j \leq i$ since $R_i = F(R_{i+1})$.

Besides, it is worth to recall that the first value of the one way chain used to generate round sequence numbers is uploaded securely into the sensors before deployment during the bootstrapping phase. □

Proposition 9 (*Authentication of sub-branch tags*). *SEIF guarantees authentication of sub-branch tags. In other words, an intruder cannot forge valid tags on behalf of the sink.*

Proof. When the sink starts a new round, it distributes to the sub-roots their respective tags, which represents the next unrevealed value of distinct OHC. This transfer is accomplished via root nodes through a secure tunnel. Since sensors maintain a chain verifier for every chain, they can easily check if a received *tag* was really generated by the sink by verifying the following relation:

$$\exists j,k : CV_{i,j} = F^k(tag)$$

An intruder cannot generate valid tags on behalf of the sink since it is computationally infeasible to calculate tag_{i+1} knowing tag_j with $j \leq i$ since $tag_i = F(tag_{i+1})$.

Besides, it is worth to recall that the first value of the one way chains used to generate sub-branch tags are uploaded securely into the sensors before deployment during the bootstrapping phase.

Moreover, to avoid that an intruder replays the tag of a sub-branch into another sub-branch inside the same round, the transmission of tags from the sink to the sub-roots (two hops) are encrypted using pairwise keys generated using a secure key management scheme (Xiao et al., 2007).

Proposition 10 (*Authentication of parents requests*). *SEIF guarantees authentication of route construction requests relayed by the parent nodes. In other words, an intruder cannot forge route requests on behalf legitimate nodes.*

Proof. To guarantee parent authentication, each sensor uses a local one way chain providing sequence numbers for its local broadcasts (one hop broadcasts). When a node relays a route request to its children it appends a new sequence number P_i to the request.

$$i \rightarrow * : i,n,p,V,R,P_i$$

Child nodes j verify the authenticity of the received sequence number P_i against the last sequence number: $NV_{j,i} = F(P_i)$.

Thus, an intruder cannot forge a route request on behalf of a legitimate parent. Indeed, an intruder cannot calculate the subsequent sequence numbers since it is computationally infeasible to calculate P_{i+1} knowing P_j with $j \leq i$ since $P_i = F(P_{i+1})$, where F is a one way hash function.

Besides, the first sequence number is sent to the one hop neighborhood encrypted using the local broadcast key BK_j on the parent node. This local broadcast key is established using some secure key management technique (Zhu et al., 2003; Xiao et al., 2007).

Moreover, if the intruder replays an old sequence number, the one hop neighbors of the legitimate node will reject the sequence number since the condition $\exists j,k : NV_{i,j} = F^k(P_i)$ would not be verified for all $k = 1 \ldots D$.

We also assume that if an intruder receives a message from a node i, all i's one hop neighbors receive the message too. In other words, an intruder cannot replay the current i's authenticator (the current i's OHC) to diffuse information of another valid sub-branch of the current round, because if it has received the current i's authenticator it means that all i's one hop neighbors have also received the current authenticator. Hence, they would detect the replay of the last value of the i's OHC. As far as we can say, this is not a strong assumption, since it is very hard for an intruder to prevent node's one hop neighbors from receiving a message that it has received itself. □

7. Performance evaluation and simulations

In this section, we will study the behavior of SMRP and SEIF compared to other solutions through simulations. We have implemented the protocols using the TinyOS environment (Hill et al., 1979). For a concise analysis of the energy consumption, we have used the Avrora tool (Titzer et al., 2005) that simulates and analyzes programs written for the AVR micro-controller, found in the Mica2 sensor nodes. It gives detailed reports about the energy consumption of different components, like: radio, CPU,… *etc*. To estimate the reliability and the average lifetime of the network, we used the Python Networkx library for different topology generations, routes calculation, and measuring the robustness of the constructed routing graphs with respect to the considered protocols. We studied the impact of the network topology on the robustness of the different protocols. Namely, we considered two types of topologies:

- *A random uniform topology*: in this topology, nodes are deployed uniformly on a square surface. A link exists between two nodes if their distance is less or equal to the radio range which is a parameter of the topology generation algorithm. This type of topology is commonly used to simulate WSN.
- *A scale free Barabasi Albert graph* (Barabási and Albert, 1999): in this type of networks known to be scale free, the distribution of links follows a power law. Each new node is connected to the existing topology through m links which is a parameter of the graph. This privileges the emergence of some nodes with high degrees and most of the others with relatively low degrees. This type of graphs is largely used to model computer networks, Internet, WWW, etc.

In addition to our solutions, we have also implemented a variety of existing protocols representing different routing approaches:

- SeRINS is a secure and non-disjoint multipath protocol.
- EINSENS is a secure and single-path protocol.
- H-SPREAD is a non-secure and node-disjoint multipath protocol.
- H-SPREAD Basic is a non-secure and node-disjoint multipath protocol limited to the first phase of the H-SPREAD protocol using one broadcast per node.
- TinyOS beaconing is a non-secure and single-path protocol.

For the family of secure protocols, we have used the TinySec library (Karlof et al., 2004) for all cryptographic operations, such as encryption and hash functions.

7.1. Mean time to failure

The mean time to failure (MTTF) represents an important metric to estimate the contribution of a solution to improve the network lifetime. It is defined as the average period of time during which a system is considered functional and can deliver sensed data to the sink. Applying this definition, we have considered that a routing topology is not functional when some sensors become incapable of reaching the sink. At this time, a reconstruction of the communication topology is necessary to repair the system. Thereby, the MTTF gives also an estimation of the required interval between two tree constructions. This estimation represents a precise information to network designers for establishing an optimal schedule of topology creation.

To evaluate this metric, we have simulated the protocols using Networkx library to obtain the constructed routing topologies while considering two deployment scenarios: uniform deployment over a square surface and a scale free Barabasi Albert power law graph. With the resulting routing topologies, we have simulated failures of nodes as a Poisson process with a rate of two failures per unit of time. When a failure occurs, we randomly select an active sensor from the network and remove it from the topology. Afterward, we verify whether the resulting graph is still connected to simulate a new failure. In the case of a disconnected graph, the system is considered "not functional" and the summation of the intervals between failures gives the time to failure. To estimate the MTTF, we considered the average of 1000 iterations for each simulation scenario and calculated the 0.96 confidence interval for each point. The confidence interval is plotted as a bar error surrounding the average value.

In a first time we fixed the average degree of the nodes and were interested in analyzing the impact of the network size on the MTTF while considering two deployment scenarios: uniform topology and scale free topology. Figure 6 presents the simulation results. We remark that the deployment scenario has a strong impact on the robustness of the considered protocols. Any way, our approach based on the concept of sub-branches depicts very good performance. When considering a scale free Barabasi Albert topology, SMRP/SEIF outperforms the other routing schemes (cf. Fig. 6(a)). This can be explained by the fact that this kind of power law links distribution tends to concentrate nodes around the sink which increases the number of sub-branches and hence the number of alternative routes per node. In the case of a uniform

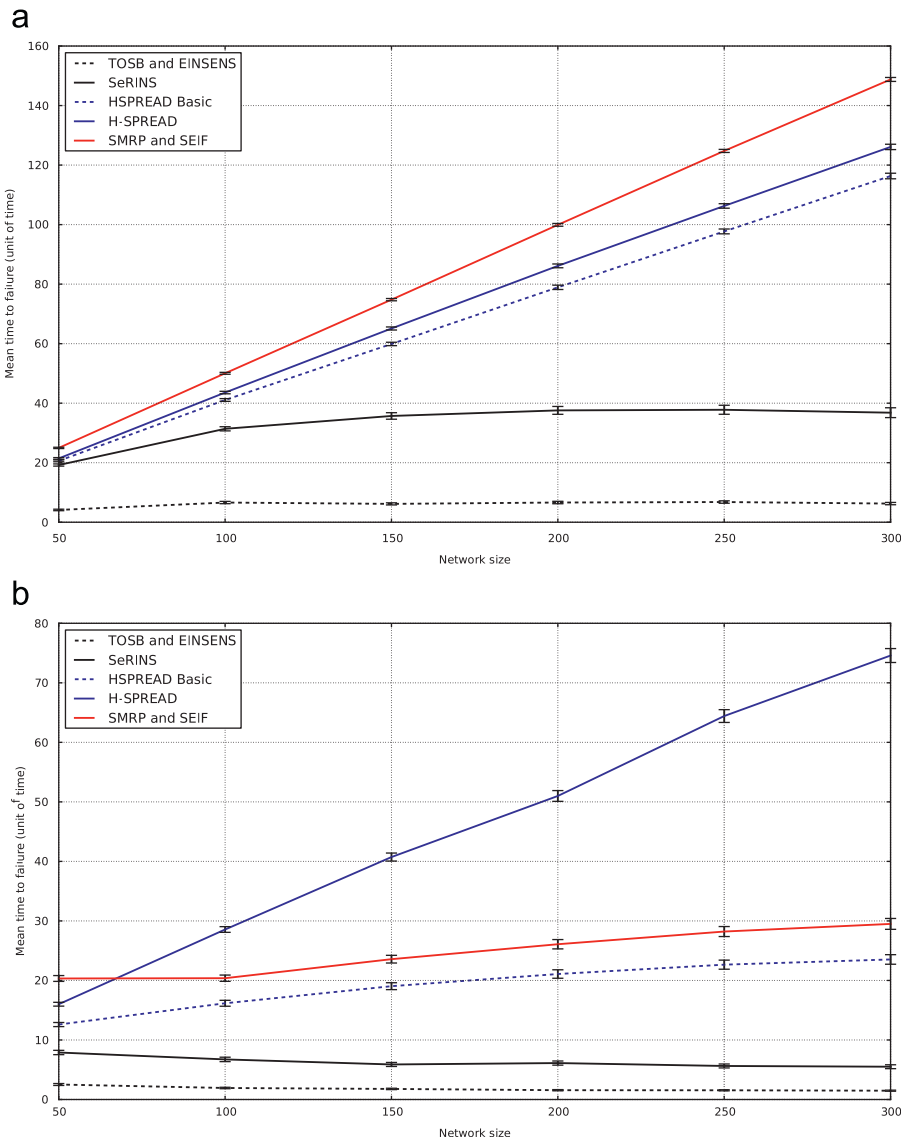

Fig. 6. Mean time to failure (MTTF) when the average degree is fixed to 20. The MTTF is the average of 1000 iterations for each point. The 0.96 confidence interval is displayed as bar errors surrounding the average MTTF values. (a) Barabasi Albert topology, (b) uniform topology.

topology (cf. Fig. 6(b)), the MTTF of SMRP/SEIF remains high but H-SPREAD depicts better performance. Indeed, in H-SPREAD nodes forward all possible disjoint routes to their neighbors. Whereas, in SMRP/SEIF only one parent chosen randomly is forwarded to child nodes. However, H-SPREAD consumes a lot of energy, due to these multiple transmissions, compared to SMRP/SEIF as we will see in the following section. Therefore, we believe that SMRP/SEIF provides a better trade-off between robustness and energy consumption overhead.

The results also demonstrate the impact of the type of redundancy on the network lifetime. Even if nondisjoint multi-path protocols, like SeRINS, offer some redundancy, they do not provide any control on its *quality*. This uncontrolled redundancy cannot improve enough the fault tolerance of the routing topology since the discovered paths tend to intersect, behaving as single-path topologies.

Then we were interested in analyzing the impact of the average degree on the MTTF. We considered a 100 nodes graphs and varied the average degree for the two types of topologies. Our point of focus was mainly to determine the threshold from which our approach outperforms the other solutions depending on the deployment scenario, and how much is realistic this degree. Figure 7 illustrates the variation of the MTTF with respect to the average degree while considering three deployment scenarios.

We notice that for a uniform topology (cf. Fig. 7(a)), H-SPREAD behaves better than the other protocols. But as explained in the following sections, it consumes a lot of energy because of the high number of messages per node required to forward all possible alternative routes. Our approach SMRP/SEIF comes in the second position and we believe this is a good trade-off since it consumes a few of energy compared to the other approaches while increasing the MTTF. We notice also that when the average degree exceeds 32% of the network size, SMRP/SEIF outperforms H-SPREAD, but this is a too high degree and hardly reached in practice. In the case of a scale free deployment (cf. Fig. 7(b)), our approach depicts the best performance for the same reasons explained above. We notice that the MTTF is the higher independently of the average degree since the distribution of the degree follows a power law which means that the density of nodes is anyway high around the sink which increases the number

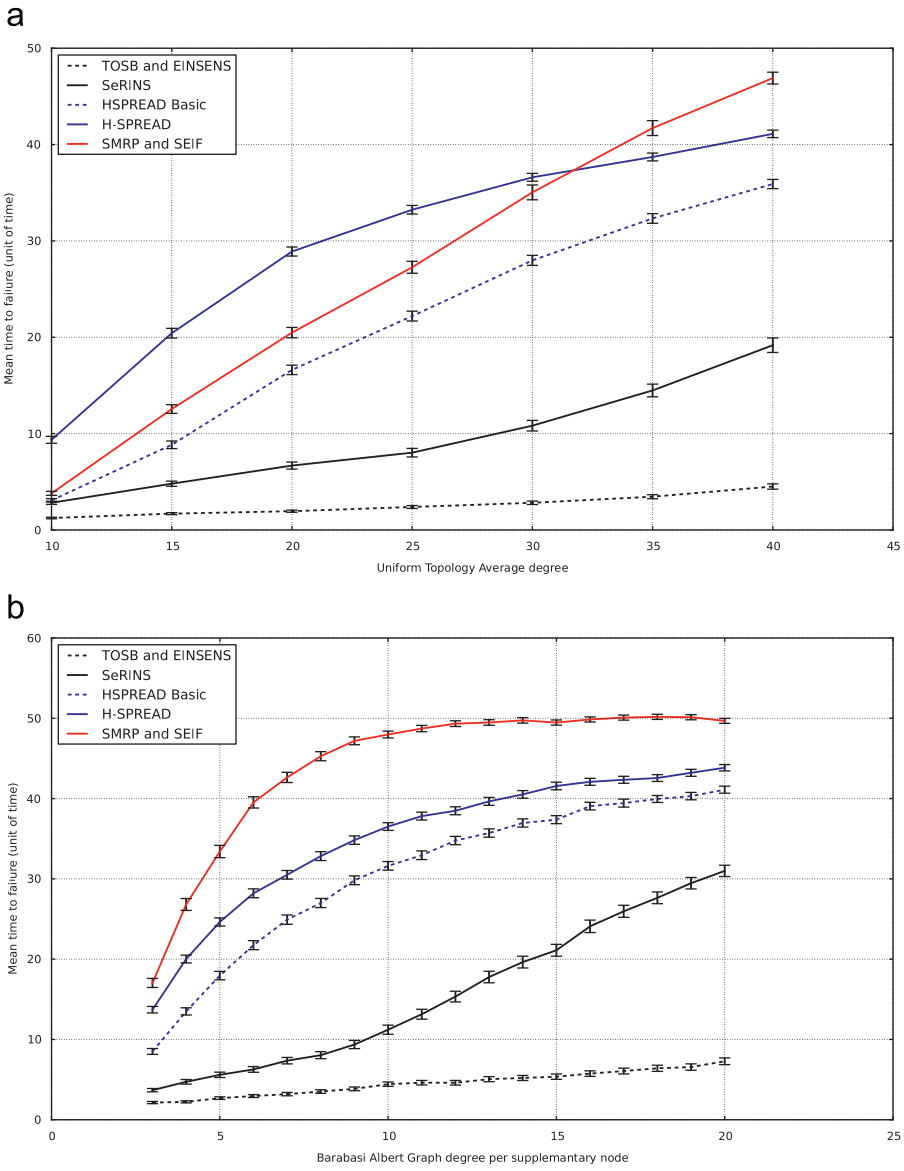

Fig. 7. Mean time to failure (MTTF) with respect to the average degree when network size is fixed to 100. The MTTF is the average of 1000 iterations for each point. The 0.96 confidence interval is displayed as bar errors surrounding the average MTTF values. (a) Uniform topology, (b) Barabasi Albert topology.

of sub-branches and hence the number of disjoint routes per node.

7.2. Energy consumption and computation overhead

Energy conservation is another compulsory goal in WSN architectures. It is not interesting to build a highly reliable or secure system that drains excessively the energy resources of sensors. One of the design goals of SMRP was to use only one message per node to conserve energy, while discovering more alternative paths.

We used Avrora simulator to evaluate the energy consumption induced by our implementation of the protocols using TinyOS. Figure 8 shows the energy consumption of the studied protocols during one round. We remark that SMRP reached the defined goal since the protocol presents near-optimal total energy consumption comparable to the simple TinyOS beaconing protocol (cf. Fig. 8(a)). In contrast, H-SPREAD generates an excessive communication overhead due to its extended branch-aware flooding that aims to discover more paths at the cost of introducing more calculations and message exchanges between sensors. Indeed, in the second phase of H-SPREAD, each node forwards all the discovered branches to its sibling and parent nodes, which increases communication overhead and energy consumption.

Studied secure protocols have globally the same performance, with a slight advantage to our protocol SEIF. Because SEIF involves several security verifications based on hash calculations, it consumes more energy than its "plain-text" version SMRP. Nevertheless, our solution induces less computation overhead compared to H-SPREAD which does not use cryptographic operations, and induces almost the same overhead as SeRINS and EINSENS with the merit to be multipath contrary to EINSENS and node-disjoint contrary to SeRINS (cf. Fig. 8(b)).

We notice that the energy consumption induced by SeRINS increases with the network size (cf. Fig. 8(a)). This can be explained by the fact that SeRINS carries out integrity verifications for each discovered route. Since routes are not node disjoint in the case of SeRINS, their number is greater. Then, the bigger is the network the greater is the number of integrity verifications and hence the greater is energy consumption. Indeed, we remark in (cf. Fig. 8(b)) that the CPU energy consumption increases with the network size. Moreover, SeRINS should require more energy in presence of intruders due to its *hybrid approach*, which will be explained in the next section.

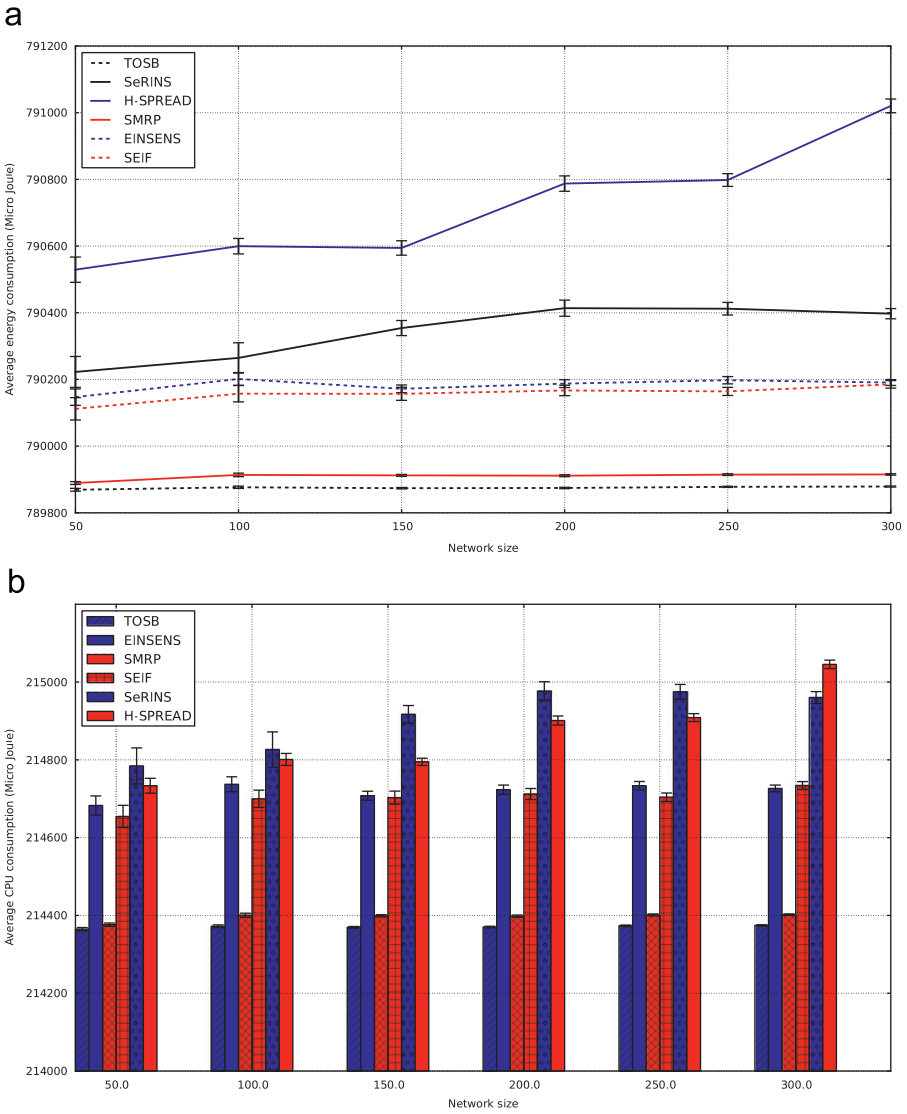

Fig. 8. Average energy consumption when the average degree is equal to 20. (a) Total energy, (b) CPU consumption.

7.3. Detection overhead

One of the main features of SEIF is its in-network verification. Sensors rely only on local information to successfully detect forged routing messages. Therefore, any intrusion attempt is instantly detected without additional delay. The same property is found in EINSENS, since it is also a totally distributed protocol. In contrast, SeRINS is a hybrid protocol in which sensors can perform only partial verifications that limit the ability of sensors to make local decisions in presence of suspect messages. Indeed, when a sensor detects a suspicious packet, it alarms the sink which must collect more information on the suspect node from its neighbors. This process is achieved via successive broadcasts, which is too expensive in large networks causing additional delay and overhead to detect the intruder.

7.4. Resiliency and fault tolerance

Even with various security countermeasures and multipath routing, a WSN is not totally immune from intruder penetrations and failures. To evaluate the capability of secure multipath protocols to tolerate the presence of intruders and failures, we have measured the resiliency of the different schemes which is the ratio of the minimum number of compromised and/or failed nodes that can disconnect the constructed topology making sensors unable to reach the sink node.

In a first time, we were interested in analyzing the impact of the network size on the resiliency of the protocols while considering tow deployment scenarios: uniform topology and scale free topology (Barabsi Albert graphs). To estimate the resiliency, we have carried out 1000 iteration for each simulation scenario and considered the average as the value of the resiliency. For each point, we calculated the 0.96 confidence interval that is plotted as a bar error surrounding the average value. The results shown in Fig. 9 confirm the conclusions drawn from the study of the MTTF metric. Our approach SMRP/SEIF depicts the better trade-off between resiliency and energy consumption overhead when considering an uniform topology (cf. Fig. 9(a)). Even though the resiliency of SMRP/SEIF decreases beneath the resiliency of H-SPREAD while increasing the network size, the latter consumes a lot of energy which would drain the nodes' batteries earlier than in the case of SMRP/SEIF. Obviously, with a scale free deployment

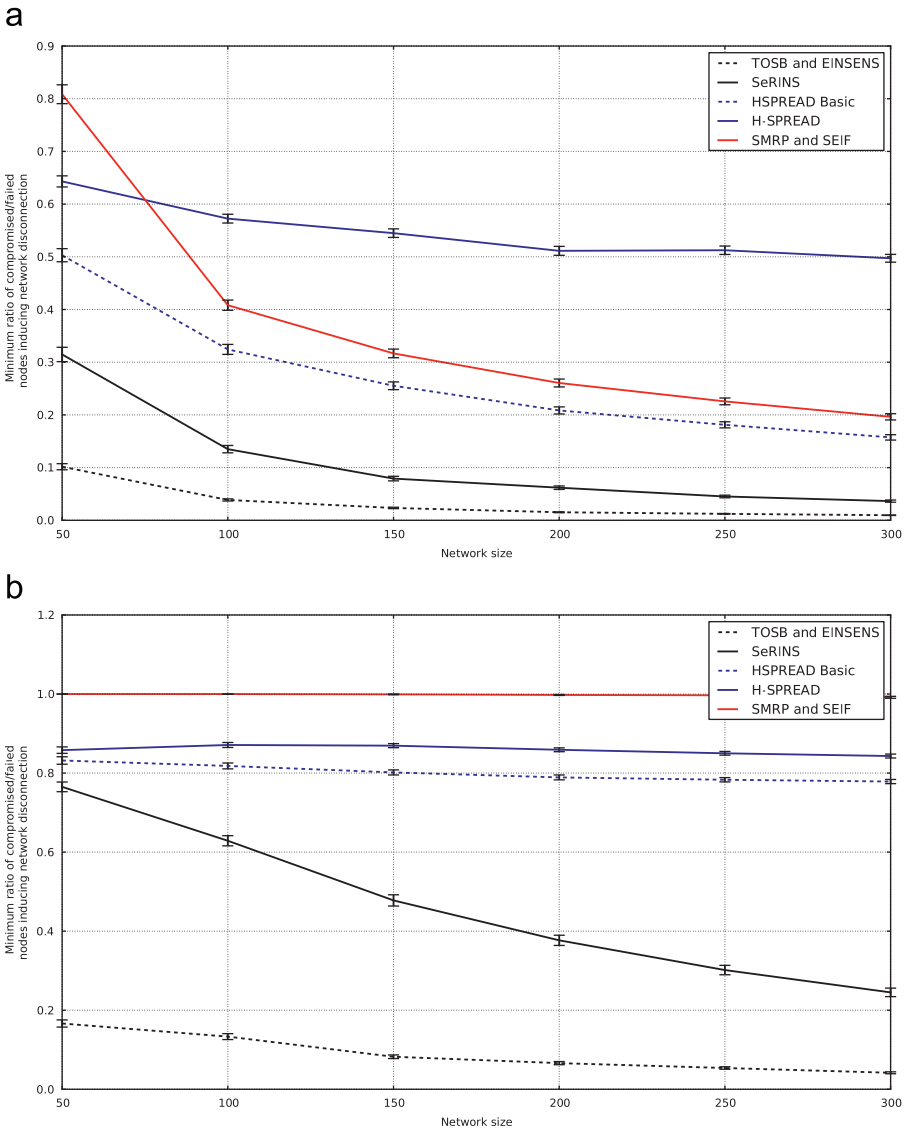

Fig. 9. Resiliency of the routing topology when the average density of sensors is equal to 20. The 0.96 confidence interval is plotted as a error bar surrounding the average value. (a) Uniform topology, (b) Barabasi Albert topology.

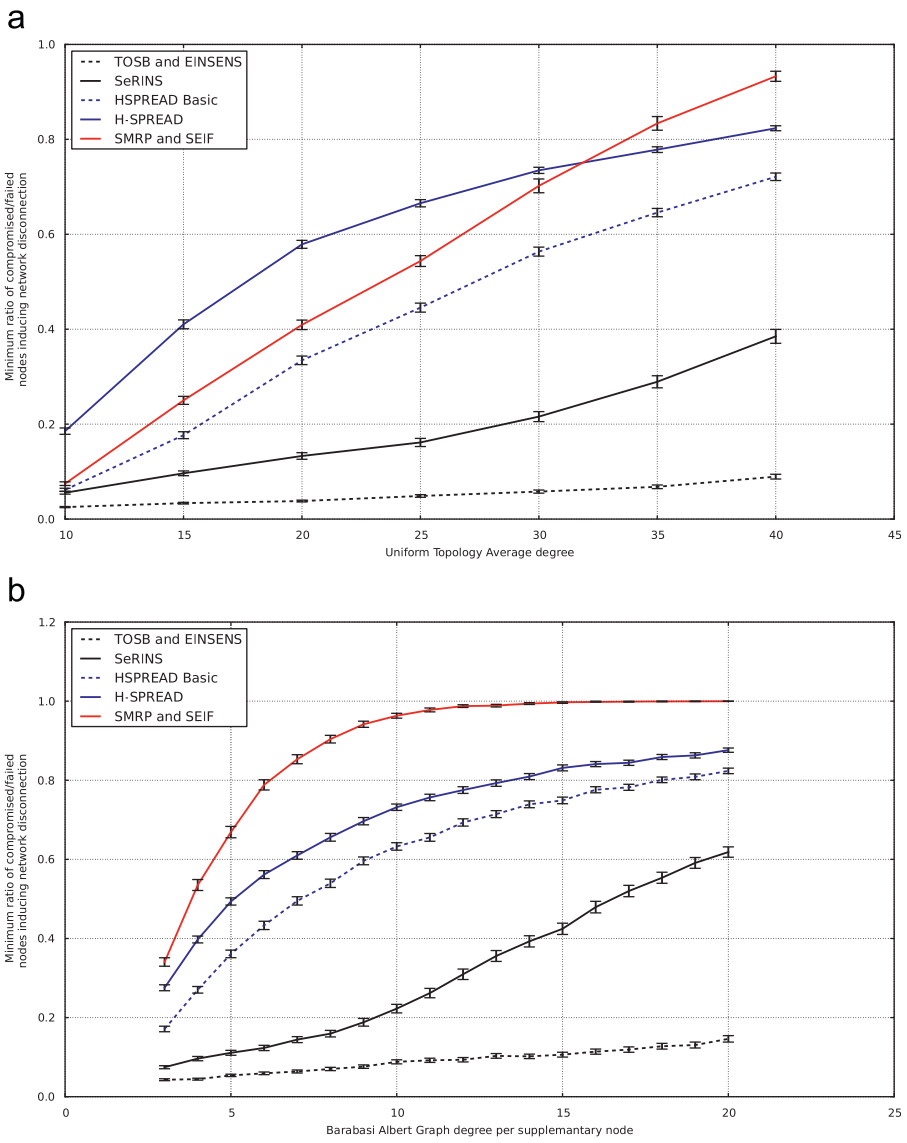

Fig. 10. Resiliency with respect to the average degree when network size is fixed to 100. The resiliency is the average of 1000 iterations for each point. The 0.96 confidence interval is displayed as bar errors surrounding the average resiliency values. (a) Uniform topology, (b) Barabasi Albert topology.

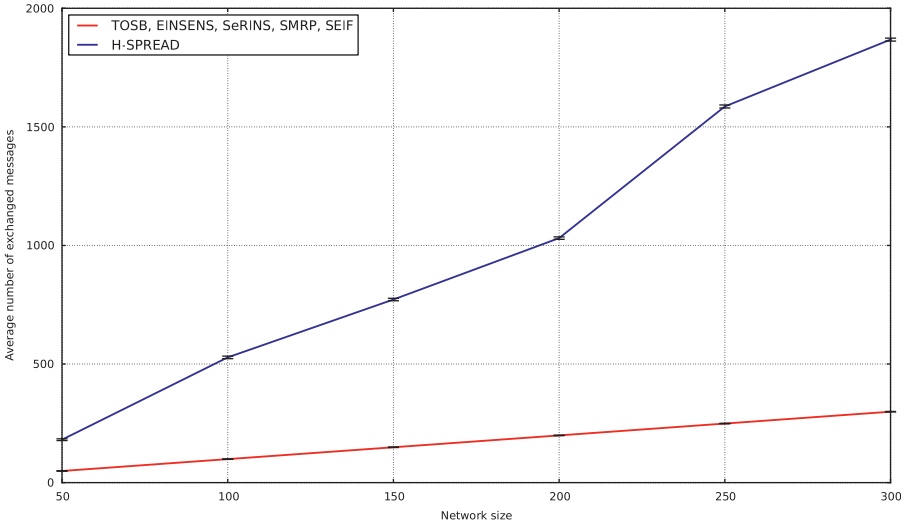

Fig. 11. Number of exchanged messages during one round.

(cf. Fig. 9(b)), SMRP/SEIF provide the highest resiliency since the number of sub-branches would increase.

The TinyOS beaconing routing scheme and EINSENS depict the lowest performance because of their single-path nature. SeRINS provides a low resiliency too because the constructed multiple routes are not node-disjoint.

Then we were interested in the impact of the average network degree on the resiliency of the routing schemes while considering the different deployment scenarios. Again the results in Fig. 10 confirm our analysis of the MTTF metric results.

In the case of a scale free deployment (cf. Fig. 10(b)), our approach depicts the best performance. We notice that the resiliency is the higher independently of the average degree since the distribution of the degree follows a power law which means that the density of nodes is anyway high around the sink.

Our approach provides also a good trade-off when it comes to uniform topologies (cf. Fig. 10(a)). The resiliency is the highest with H-SPREAD, but the latter consumes a lot of energy because of the high number of messages per node required to forward all possible alternative routes.

7.5. Scalability and bandwidth overhead

To guarantee large scale deployments of WSN, it is important to study the scalability of the proposed solutions and the required number of control messages and hence the induced bandwidth overhead. Protocols requiring feedbacks from sensor nodes, such as INSENS, suffer from a poor scalability. This is due to the one-to-one dialog between each sensor and the base station, which leads to an excessive bandwidth overhead when dealing with a large number of sensors. By removing the feedback phase, tree-based protocols (like EINSENS, SEIF and SeRINS) improve significantly the overall scalability by requiring only one message per sensor in order to establish the communication topology. H-SPREAD requires too much message exchanges in order to forward all discovered alternative routes. These multiple transmissions induce a high bandwidth overhead, collisions, and energy depletion. Figure 11 illustrate the evolution of the number of exchanged messages in the network while increasing the network size. We notice that TOSB, SeRINS, EINSENS, SMRP and SEIF scale well since they require only one message per node to establish the routes. In contrast, the number of messages per node in the case of H-SPREAD is proportional to the network size, which means that it does not scale well with large groups.

8. Practical considerations and implementation issues

The above discussions demonstrate that our solution SEIF outperforms some representative solutions of the literature, especially in what relates to the mean time to failure, resilience against node capture, scalability, energy consumption, etc. This performance can be achieved under some practical constraints that can be summarized in the following points:

- Our solution relies on the construction of sub-branch disjoint paths, which means that intersections may occur at root nodes (neighbors of the sink). This assumes, as stated in Section 3.3 that root nodes should be powerful, and ideally well protected. This is what we called the security perimeter.
- Our solution, and multipath routing solutions generally, aim to create a maximum number of node-disjoint routes. Following our simulations, we concluded that route disjointness is possible only in dense networks. Otherwise, the number of node-disjoint routes would be low.
- The secure version of our solution provides branch authentication using one way hash chains. A different hash chain should be attributed to each root node. Therefore, the maximum number of root nodes should be known before deployment to allow the generation of enough distinct hash chains.

9. Conclusion

In this paper, we have investigated the problems of fault tolerance and intrusion tolerance. These two concepts represent important issues in WSN. Existing solutions addressing these problems suffer from a poor trade-off between the offered fault tolerance degree and the induced construction overhead. To achieve such trade-off, we propose SEIF, an intrusion-fault tolerant routing scheme offering a high level of reliability through a secure multipath communication topology. SEIF relies on one way hash chains to secure the construction of a multipath many-to-one dissemination topology. One way hash chains guarantee authentication of exchanged control messages without incurring high energy consumption. We have analyzed the disconnection probability of multipath topologies built using our sub-branch disjoint approach with respect to different data forwarding schemes. We showed that our scheme enhances the robustness of the created multipath topologies under the reasonable assumption of the existence of a reliability perimeter around the bases station. Furthermore, simulation results using TinyOS and topology analysis using Networkx library show the enhancements of our scheme SMRP/SEIF over other solutions of the literature with regard to some important metrics, such as: the mean time to failure, energy consumption, resiliency and some security attacks detection overhead.

Appendix A. Proofs

Proof. (proposition 1) In the case of full duplication over node-disjoint multipath construction, a node is disconnected from the sink if all available routes are disconnected from the sink:

$$f_i = Pr[\forall \pi \in \{\pi_i\} : F(\pi)] = \prod_{\pi \in \{\pi_i\}} Pr[F()\pi] = \prod_{\pi \in \{\pi_i\}} (1-(1-\alpha)^{|\pi|}) \qquad \square$$

Proof. (proposition 2) In the case of random parent selection paradigm over node disjoint multipath construction, the probability of node i disconnection from the sink becomes

$$f_i = Pr[F(\text{selected path})]$$

Using the total probability law and considering the active parent is selected uniformly:

$$f_i = \sum_{\pi \in \{\pi_i\}} (Pr[F(\pi)/\text{choose path } \pi] \times Pr[\text{choose path } \pi])$$

$$= \sum_{\pi \in \{\pi_i\}} (1-(1-\alpha)^{|\pi|}) \times \frac{1}{|\{\pi_i\}|} \qquad \square$$

Proof. (proposition 3) In the case of (t,n)-loss tolerant duplication over a node-disjoint multipath construction, the probability that node i becomes disconnected from the sink can be calculated as follows:

$$f_i = Pr[\text{at least}(n-t+1)\text{pieces are lost}]$$

$$= \sum_{k=n-t+1}^{n} Pr[\text{exactly k pieces are lost}]$$

$$= \sum_{k=n-t+1}^{n} \left(\sum_{s \in subs(\{\pi_i\},k)} (Pr[F(s)] \times Pr[\text{no failure in} \{\pi_i\}-s]) \right)$$
$$= \sum_{k=n-t+1}^{n} \left(\sum_{s \in subs(\{\pi_i\},k)} \left(\prod_{\pi \in s} (1-(1-\alpha)^{|\pi|}) \times \prod_{\pi \in \pi_i - s} (1-\alpha)^{|\pi|} \right) \right) \quad \square$$

Proof. (proposition 4) In the case of full duplication over a sub-branch disjoint multipath construction, the probability that node i becomes disconnected from the sink can be calculated as follows:

$$f_i = Pr[\forall \pi \in \{\pi_i\} : F(\pi)]$$
$$= Pr\left[\bigwedge_{\pi \in \{\pi_i\}} F(\pi) \right]$$
$$= \prod_{r \in roots_i} Pr\left[\bigwedge_{\pi \in \{\pi_i(r)\}} F(\pi) \right]$$
$$= \prod_{r \in roots_i} Pr[\text{r has failed}] + Pr[\text{r has not failed}] \times Pr\left[\bigwedge_{\pi \in \{\pi_i(r)\}} F(\pi) \right]$$
$$= \prod_{r \in roots_i} (\alpha_r + (1-\alpha_r) \times \prod_{\pi \in \{\pi_i(r)\}} (1-(1-\alpha)^{|\pi|-1})) \quad \square$$

Proof. (proposition 5) In the case of random parent selection paradigm over sub-branch disjoint multipath construction, the probability that node i becomes disconnected from the sink can be calculated as follows:

$$f_i = Pr[F(\text{selected path})]$$

Using the total probability law and considering the active parent is selected uniformly:

$$f_i = \sum_{\pi \in \{\pi_i\}} (Pr[F(\pi)/\text{choose path } \pi] \times Pr[\text{choose path } \pi])$$
$$= \sum_{\pi \in \{\pi_i\}} (1-\alpha_r)(1-(1-\alpha)^{|\pi|-1}) \times \frac{1}{|\{\pi_i\}|} \quad \square$$

Proof. (proposition 6) In the case of (t,n)-loss tolerant duplication over a sub-branch disjoint multipath construction, the probability that node i becomes disconnected from the sink can be calculated as follows:

$$f_i = \sum_{k=n-t+1}^{n} \sum_{s \in subs(\{\pi_i\},k)} Pr[\forall \pi \in s : F(\pi) \text{ and } \forall \pi \in \{\pi_i\}-s : \overline{F(\pi)}]$$
$$= \sum_{k=n-t+1}^{n} \sum_{s \in subs(\{\pi_i\},k)} Pr\left[\left(\bigwedge_{\pi \in s} F(\pi) \right) \bigwedge \left(\bigwedge_{\pi \in \{\pi_i\}-s} \overline{F(\pi)} \right) \right]$$
$$= \sum_{k=n-t+1}^{n} \sum_{s \in subs(\{\pi_i\},k)} Pr\left[\bigwedge_{r \in roots_i} \left(\bigwedge_{\pi \in s \cap \{\pi_i(r)\}} F(\pi) \bigwedge \bigwedge_{\pi \in \{\pi_i(r)\}-s} \overline{F(\pi)} \right) \right]$$
$$= \sum_{k=n-t+1}^{n} \sum_{s \in subs(\{\pi_i\},k)} \prod_{r \in roots_i} Pr\left[\bigwedge_{\pi \in s \cap \{\pi_i(r)\}} F(\pi) \bigwedge \bigwedge_{\pi \in \{\pi_i(r)\}-s} \overline{F(\pi)} \right]$$
$$= \sum_{k=n-t+1}^{n} \left(\sum_{s \in subs(\{\pi_i\},k)} \prod_{r \in roots_i} G_i(s,r) \right)$$

$G_i(s,r)$ denotes the probability of disconnection within the branch identified by r, when the set s is selected as disjoint sub-branch paths. Each branch r can contain paths that belong to s or not. We can distinguish between two cases:

If all paths belonging to the branch r belong also to s, we can apply the formula of loss in the case of full duplication, since this branch has to be totally disconnected in order to satisfy that exactly k paths are disconnected.

In the case the branch r contains paths not belonging to s, the root r cannot be considered failed in the calculation of $G_i(s,r)$, since the branch r would contain in this case some paths not selected and hence not failed:

$$G_i(s,r) = Pr\left[\left(\bigwedge_{\pi \in s \cap \{\pi_i(r)\}} F(\pi) \right) \bigwedge \left(\bigwedge_{\pi \in \{\pi_i(r)\}-s} \overline{F(\pi)} \right) \right]$$
$$= (1-\alpha_r) \times Pr\left[\left(\bigwedge_{\pi \in s \cap \{\pi_i(r)\}} F(\pi) \right) \bigwedge \left(\bigwedge_{\pi \in \{\pi_i(r)\}-s} \overline{F(\pi)} \right) \right]$$
$$= (1-\alpha_r) \times \prod_{\pi \in \{\pi_i(r)\} \cap s} (1-(1-\alpha)^{|\pi|-1}) \prod_{\pi \in \{\pi_i(r)-s\}} (1-\alpha)^{|\pi|-1}$$

Finally we can write

$$f_i = \sum_{k=n-t+1}^{n} \left(\sum_{s \in subs(\{\pi_i\},k)} \prod_{r \in roots_i} G_i(s,r) \right)$$

where

$$G_i(s,r) = \begin{cases} \alpha_r + (1-\alpha_r) \times \prod_{\pi \in \{\pi_i(r)\}} (1-(1-\alpha_r)^{|\pi|-1}), \\ \text{if} \{\pi_i(r)\}-s = \emptyset \\ (1-\alpha_r) \times \prod_{\pi \in \{\pi_i(r)\} \cap s} (1-(1-\alpha_r)^{|\pi|-1}) \times \\ \prod_{\pi \in \{\pi_i(r)\}-s} (1-\alpha_r)^{|\pi|-1}, \text{if} \{\pi_i(r)\}-s \neq \emptyset \end{cases} \quad \square$$

References

Al-Karaki J, Kamal A. Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications 2004;11:6–28.

Barabási A-L, Albert R. Emergence of scaling in random networks. Science 1999;286:509–12.

Chen L, Leneutre J. On multipath routing in multihop wireless networks: security performance and their tradeoff. EURASIP Journal of Wireless Communication and Networking 2009;2009:1–13.

Deng J, Han R, Mishra S. INSENS: intrusion-tolerant routing for wireless sensor networks. Computer Communications 2006;29:216–30.

Djenouri D, Khelladi L, Badache AN. A survey of security issues in mobile ad hoc and sensor networks. Communications Surveys & Tutorials, IEEE 2005;7:2–28.

Ganesan D, Govindan R, Shenker S, Estrin D. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. SIGMOBILE Mobile Computing and Communication Review 2001;5:11–25.

Hill J, Szewczyk R, Woo A, Hollar S, Culler DE, Pister KSJ. System architecture directions for networked sensors. In: Proceedings of architectural support for programming languages and operating systems, p. 93–104.

Hou R, Shi H. A localized algorithm for finding disjoint paths in wireless sensor networks. IEEE Communications Letters 2006;10:807–9.

Karlof C, Sastry N, Wagner D. Tinysec: A link layer security architecture for wireless sensor networks. In: Proceedings of the second ACM conference on embedded networked sensor systems (SensSys), 2004.

Karlof C, Wagner D. Secure routing in wireless sensor networks: attacks and countermeasures. Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols 2003;1:293–315.

Kim M, Jeong E, Bang Y.-C, Hwang S, Kim B. Multipath energy-aware routing protocol in wireless sensor networks. In: 5th international conference on networked sensing systems, p. 127–30.

Lamport L. Constructing digital signatures from one-way function, Technical Report SRI-CSL-98, SRI International, 1979.

Lee SB, Choi YH. A secure alternate path routing in sensor networks. Computer Communications 2006;30:153–65.

Li S, Neelisetti R, Liu C, Lim A. Efficient multi-path protocol for wireless sensor networks. International Journal of Wireless and Mobile Networks 2010;2(1):110–30.

Li S, Wu Z. Node-disjoint parallel multi-path routing in wireless sensor networks. In: Proceedings of the second international conference on embedded software and systems (IEEE); 2006. p. 432–7.

Lou W, Kwon Y. H-SPREAD: a hybrid multipath scheme for secure and reliable data collection in wireless sensor networks. IEEE Transactions on Vehicular Technology 2006;55:1320–30.

Nasser N, Chen Y. Secure multipath routing protocol for wireless sensor networks. In: International conference on distributed computing systems workshops; 2007. p. 12.

Ouadjaout A, Challal Y, Bachir A, Lasla N, Bagaa M, Khelladi L. Handbook/encyclopedia on ad hoc and ubiquitous computing, World Scientific; 2009. p. 427–72.

Philip L, Nelson L, Matt W, David C. TOSSIM: accurate and scalable simulation of entire tinyos applications. In: Proceedings of the first ACM conference on embedded networked sensor systems, SenSys, 2003.

Rabin MO. Efficient dispersal of information for security load balancing and fault tolerance. Journal of the ACM 1989;36:335–48.

Sohrabi K, Gao J, Ailawadhi V, Pottie GJ. Protocols for self-organization of a wireless sensor network. IEEE Personal Communications 2000;7:16–27.

Stavrou E, Pitsillides A. A survey on secure multipath routing protocols in wsns. Computer Networks 2010;54(13):2215–38.

Titzer BL, Lee DK, Palsberg J. Avrora: scalable sensor network simulation with precise timing. In: Proceedings of the 4th international symposium on information processing in sensor networks (IPSN), Piscataway, NJ, USA; 2005. p. 477–82.

Wang L, Ma J, Wang C, Kot A. Fault and intrusion tolerance of wireless sensor networks. In: Proceedings of the 20th international parallel and distributed processing symposium; 2006. p. 7.

Xiao Y, Rayi VK, Sun B, Du X, Hu F, Galloway M. A survey of key management schemes in wireless sensor networks. Computer Communications 2007;30:2314–41.

Xiuli R, Haibin Y. A novel multipath disjoint routing to support ad hoc wireless sensor networks. In: Proceedings of the ninth IEEE international symposium on object and component-oriented real-time distributed computing (ISORC 06), Washington, DC, USA; 2006. p. 174–8.

Yang Y, Zhong C, Sun Y, Yang J. Network coding based reliable disjoint and braided multipath routing for sensor networks. Journal of Network and Computer Applications 2010;33:422–32.

Zhu S, Setia S, Jajodia S. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In: Proceedings of ACM CCS; 2003. p. 308–9.