SEIF: Secure and Efficient Intrusion Fault tolerant protocol for Wireless Sensor Networks

Abdelraouf Ouadjaout, Yacine Challal, Noureddine Lasla, Mouloud Bagaa

SEIF: Secure and Efficient Intrusion-Fault tolerant routing protocol for wireless sensor networks

Abdelraouf Ouadjaout*, Yacine Challal§, Noureddine Lasla‡, Miloud Bagaa*
*University of Science and Technology Houari Boumediene, LSI lab
Algiers, Algeria
Email: {ouadjaout,bagmoul}@gmail.com
§Compiegne University of Technology, Heudiasyc lab
Compiegne, France
Email: ychallal@hds.utc.fr
‡Institut National de formation en Informatique
Algiers, Algeria
Email: lnoureddine4@gmail.com

***Abstract*—In wireless sensor networks, reliability represents 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 multi-path communication topology. 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 multi-path selection scheme seeking to enhance the tolerance of the network and conserve the energy of sensors. Extensive simulations with TinyOS showed that our approach improves the overall Mean Time To Failure (MTTF) while conserving the energy resources of sensors.**

***Index Terms*—Sensor network, Security, Intrusion tolerance, Fault tolerance, Secure routing.**

I. Introduction

Wireless Sensor Networks (WSN) represent 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. Sensors 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 [1]. The concept behind this family of protocols is to postpone nodes failure as far as possible, but this method is not sufficient enough since these failures are *inevitable*.

More elaborate solutions consider node failure as *a normal property of the network* and enhance the network lifetime by providing tolerant mechanisms that guarantee normal operation of the network in presence of failures. Major tolerant solutions for WSN and MANET are based on the multi-path routing paradigm, which provides each sensor with alternative paths. Different kinds of multi-path schemes have been proposed, offering different levels of reliability and fault tolerance [2], [3]. 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 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 [4], [5]. 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 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 scalability suffer from poor level of fault tolerance and do not consider the intersection of built paths leading to non disjoint routes.

The contribution of this paper is twofold:

- First, we introduce a new approach of multi-path routing, called SMRP (*Sub-branch Multi-path 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 multi-path 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 II. Section III gives the design goals and a detailed description of the protocol SMRP. In section IV, we describe our secure and efficient intrusion-fault tolerant solution SEIF. Simulation results are detailed and analyzed in section V. Finally, we summarize our work and draw conclusions in section VI.

II. Related Works

Despite existing similarities between intrusion tolerance and fault tolerance design goals, they have traditionally been studied separately [6]. 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 [7]. 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 multi-path 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 [8] 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 multi-path routing by deploying several sinks and constructing a single route to each sink.

Lee et al. [9] proposed SeRINS, a secure multi-path 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 tradeoff 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.

Consequently, by analyzing both solutions, we can conclude that existing intrusion-fault tolerant approaches do not provide an acceptable tradeoff between the level of fault tolerance and the induced communication overhead.

III. SMRP

In this section, we describe our protocol SMRP, an enhancement of node disjoint path construction for one-to-many communication paradigm. In section IV, SMRP will be employed as the route selection scheme for our intrusion-fault tolerant protocol SEIF.

A. Problem definition

Redundancy represents an important concept in the design of a reliable and fault tolerant system. For that reason, node disjoint paths have been the most preferable metric in existing multi-path routing protocols. There exist different solutions for finding node disjoint paths between communicating nodes [3], [8], [10]. *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 [3]. 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 subtree 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 [3] 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.

B. 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.

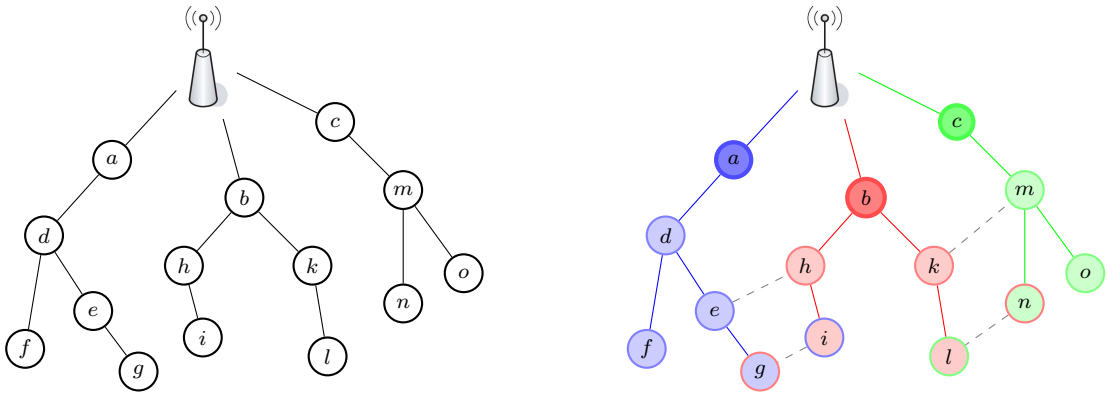


Fig. 1. The concept of branch-aware flooding. (a) A topology obtained by a simple flooding protocol like the TinyOS beaconing protocol; (b) In a branch-aware protocol, the redundant reception of construction message can be exploited to discover new paths, without adding new messages. For instance, when 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*.

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

The basic idea of SMRP follows from the following fact. Root nodes 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 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.

C. Description

The proposed method is based on the exchange of the message RREQ (*Route REQuest*) having the following format:

$$(r, parent, subBranch)$$

where :

- r : the sequence number identifying the current round.
- $parent$: the ID of the sending node.
- $subBranch$: 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.

1) Round initialization: Periodically, the sink starts the construction of a new tree by broadcasting the following message:

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

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) 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:

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

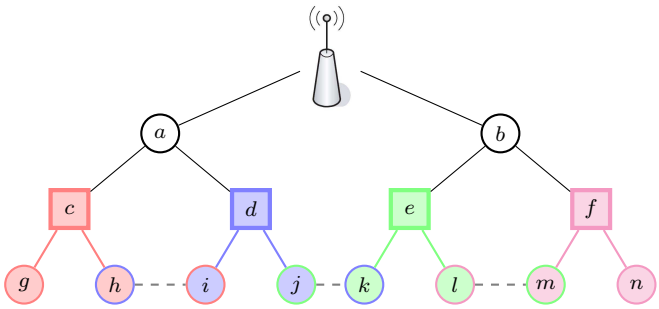


Fig. 2. Sub-branch construction of the protocol SMRP. Routes are tagged with the IDs of 2-hops neighbors of the sink. Theses nodes are named sub-roots and represented by the square nodes. When two nodes advertise two distinct sub-branches, they become *cousin*. In this example, we can distinguish between 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.

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

IV. SEIF

The protocol SEIF represents a merge between the multi-path topology offered by SMRP and an efficient in-network sub-branch authentication. This merge brings up a highly reliable and secure routing system tolerant to failures and attacks.

A. 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 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 countermeasures to verify 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 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 [8], [4].

B. Protocol overview

Instead of identifying sub-branches with simple node IDs that can be manipulated by any intruder, we have designed a solution based on the concept of one-way hash chains (OHC) [11] that prevents intruders from advertising inexistent sub-branches. A one-way hash chain is a sequence of numbers $(K_i)_{0\leq i\leq n}$ generated by a one-way function F as follows:

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

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}. 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 j^{th} 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*. 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:

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

Since these tags are delivered solely by the sink node, the latter can *control* the number of authentic sub-branches during the current round. This centralized distribution blocks any attempts of malicious nodes to falsify the correct routing view.

For the verification of new round initialization, we must use another OHC as a one way sequence number for rounds. Therefore, only the sink can provide the correct next sequence number to launch a new construction round.

To guarantee parent authentication, each sensor must have a local chain providing sequence numbers for its local broadcasts. Contrary to previous OHCs used for multi hop authentication, this type of chain is necessary for *one hop authentication* and each sensor possesses its own chain stocked locally. However, this authentication is not sufficient to counter hello flooding attacks. This type of attacks necessitates discovering and authenticating the reachable neighborhood. Our solution is based on combining an OHC-based authentication with some key management techniques.

Since many key management protocols establish the key materials using *challenge-response mechanisms* [12], [13], two sensors will not share a secret key only if they have a bidirectional link. Knowing that OHC-based authentication can not be done without initializing a verifier with an adequate value of the chain, a sensor will send the first unused value of its local chain encrypted with its broadcast key. This way, only reachable neighbors will decrypt the message and initialize their chain verifier corresponding to the sending neighbor, in order to authenticate its future messages.

C. Detailed description

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:

$$i \rightarrow * : i, E(BK_i, V) \tag{1}$$

  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.

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:

$$\begin{aligned} sink \rightarrow i : \quad & E(K_{sink,i}, subRoot_1 || \ n_1 \ || \ p_1 \ || \ V_1 \\ & || \ ... \ || \\ & subRoot_m || \ n_m \ || \ p_m \ || V_m \ || \ R) \end{aligned} \tag{2}$$

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 (2), 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 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:

$$i \rightarrow subRoot_k : E(K_{i,subRoot_k}, n_k \ || \ p_k \ || \ V_k \ ||R) \tag{3}$$

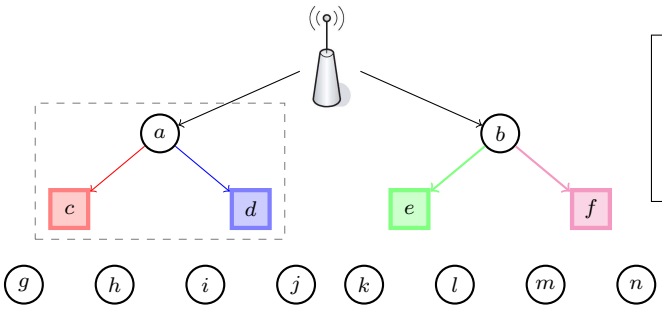
After a sensor decrypts the message (3) and verifies the round and sub-branch sequence numbers (using the same procedures as described above), it can start the creation of its own sub-branch. Using the provided tag, it can now pretend to be a sub-root for the current round. An example describing the different steps during the tag distribution phase is presented in Fig. 3(a).

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

$$i \rightarrow * : i, n, p, V, R, P_i \tag{4}$$

where :

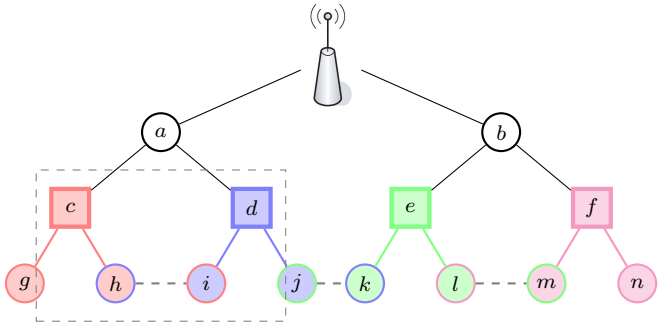
[1]Because SEIF is independent from deployment, the chains are not intrinsically linked to sub-root nodes. From a round to another, the affectation of chains to these nodes can change without disturbing the execution of the protocol.



$sink \rightarrow a : E(K_{sink,a}, c||n_1||p_1||V_1||d||n_2||p_2||V_2||R)$

$a \rightarrow c : E(K_{a,c}, n_1||p_1||V_1||R)$
$a \rightarrow d : E(K_{a,d}, n_2||p_2||V_2||R)$

(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 verify 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.



$c \rightarrow * : c, n_1, p_1, V_1, R, P_c$
$d \rightarrow * : d, n_2, p_2, V_2, R, P_d$

$h \rightarrow * : h, n_1, p_1, V_1, R, P_h$
$i \rightarrow * : i, n_2, p_2, V_2, R, P_i$

(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 broadcasted by c, h verifies the following equations : (1) $NV_{h,c} = F(P_c)$, (2) $RV_h = F(R)$ or $R = RV_h$, (3) $CV_{h,n_1} = F^{p_1 - P_{h,n_1}}(V_1)$. When two neighbors advertise two distinct valid tags, they become *cousins*.

Fig. 3. An example of the secure sub-branch aware flooding provided by SEIF.

- n, p ,V and R represent the values received from the root node within the message (3).
- P_i is the first unused value of the local OHC for one hop authentication.

When a sensor j receives the message (4), it authenticates the sending node by verifying if P_i represent 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 (2) and (3), 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 update $CV_{j,n}$ 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 wether 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 (4) using the sub-branch tag of the chosen main parent. Fig. 3(b) gives an example of tree construction and alternative path discovery in the protocol SEIF.

V. Simulations and Analysis

In this section, we will study the behaviour of SMRP and SEIF through simulation results and theorycal analysis. We have implemented SMRP and SEIF using the TinyOS environment [14]. We carried out the simulations using two tools. To estimate the reliability and the average lifetime of the network, we have used the TOSSIM simulator that ships with the TinyOS environment [15]. For a concise analysis of the energy consumption, we have used the Avrora tool [16] that simulates and analyzes programs written for the AVR microcontroller, found in the Mica2 sensor nodes. It gives detailed reports about the energy consumption of different

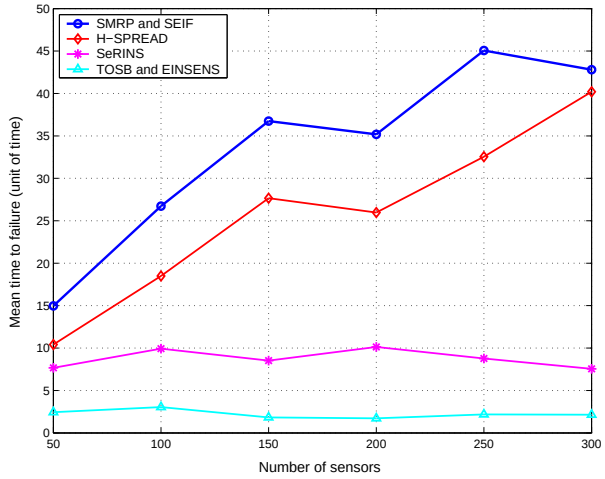


Fig. 4. Mean Time To Failure (MTTF) when the average density is fixed to 20

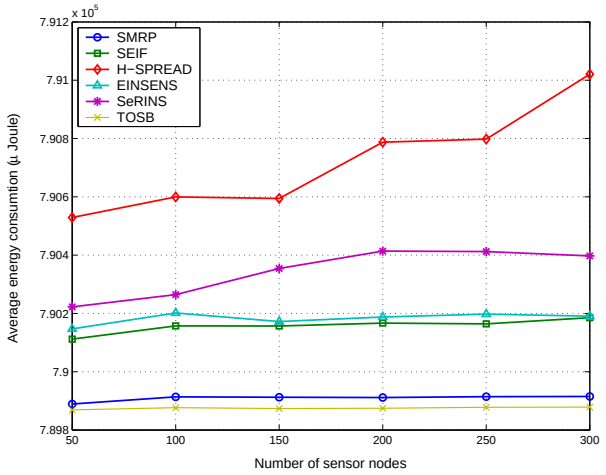


Fig. 5. Average energy consumption when the average density is equal to 20.

components, like : radio, CPU, ... *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 multi-path protocol.
- EINSENS is a secure and single-path protocol.
- H-SPREAD is a non-secure and node-disjoint multi-path protocol.
- TinyOS beaconing is a non-secure and single-path protocol.

For the family of secure protocols, we have used the TinySec library [17] for all cryptographic operations, such as encryption and hash functions.

A. 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 represent a precise information to network administrators for establishing an optimal schedule of topology creation.

To evaluate this metric, we have simulated the protocols using TOSSIM to obtain the constructed routing topologies. With these topologies, we have simulated failures of nodes as a Poisson process with a rate of 2 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 have executed 200 simulations for each scenario.

Fig. 4 presents the simulation results of the MTTF metric. We remark that our approach based on the concept of sub-branches outperforms the other routing schemes, including node disjoint multi-path. This can be explained by the fact that node disjoint routes are more difficult to find and less abundant because they obey to stricter restrictions. The results also demonstrates the impact of the type of redundancy on the network lifetime. Even if non disjoint multi-path protocols, like SeRINS, offer some redundancy, they don’t provide any control on its *quality*. This uncontrolled redundancy can not improve enough the fault tolerance of the routing topology since the discovered paths tend to intersect, behaving as single path topologies.

B. Energy consumption

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.

Fig. 5 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 energy consumption comparable to the simple TinyOS beaconing protocol. In contrast, H-SPREAD generated an excessive communication overhead due to its extended branch-aware flooding that aims to discover more paths at cost of introducing more message exchange between sensors.

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, SeRINS should require more energy in presence of intruders due to its *hybrid approach*, which will be explained in the next section.

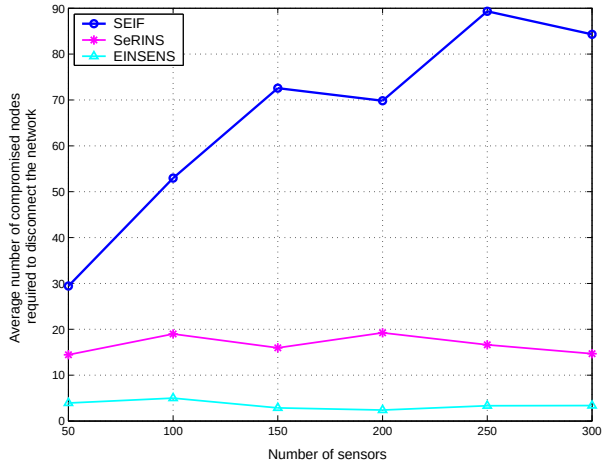


Fig. 6. Resiliency of the routing topology when the average density of sensors is equal to 20.

C. 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.

D. Resiliency against the presence of intruders

Even with various security countermeasures, a WSN is not totally immune from intruder penetrations. To evaluate the capability of secure protocols to tolerate the presence of intruders, we have measured the minimum number of compromised nodes that can disconnect the constructed topology making sensors unable to reach the sink node. For each scenario, we have carried out 200 simulations. The results shown in Fig. 6 confirms the conclusions drawn from the study of the MTTF metric. Among studied secure protocols, the protocol SEIF offers the best resiliency and enhances significantly the survivability of the network. Moreover, the resiliency of SEIF scales well with network growth : the bigger is the network, the higher is the tolerance. In contrast, EINSENS and SeRINS provide a constant tolerance that is not affected by the number of sensors in the network.

VI. 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 tradeoff between the offered tolerance and the induced construction overhead. To achieve such tradeoff, we propose SEIF, an intrusion-fault tolerant routing scheme offering a high level of reliability through a secure multi-path communication topology. SEIF relies on one way hash chains to secure the construction of a multi-path many-to-one dissemination tree. One way hash chains guarantee authentication of exchanged control messages without incurring high energy consumption. Furthermore, simulation results using TinyOS show that the Mean Time To Failure (MTTF) of our solution SEIF exceeds the MTTF of representative solutions in the literature.

References

[1] J. Al-Karaki and A. Kamal, “Routing techniques in wireless sensor networks: a survey,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6 – 28, 2004.

[2] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, “Highly-resilient, energy-efficient multipath routing in wireless sensor networks,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 4, pp. 11–25, 2001.

[3] W. Lou and Y. Kwon, “H-SPREAD: a Hybrid Multipath Scheme for Secure and Reliable Data Collection in Wireless Sensor Networks,” *IEEE Transactions on Vehicular Technology*, vol. 55, no. 4, pp. 1320–1330, 2006.

[4] C. Karlof and D. Wagner, “Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures,” *Elsevier’s AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, vol. 1, no. 2–3, pp. 293–315, September 2003.

[5] D. Djenouri, L. Khelladi, and A. N. Badache, “A survey of security issues in mobile ad hoc and sensor networks,” *Communications Surveys & Tutorials, IEEE*, vol. 7, no. 4, pp. 2–28, 2005.

[6] L. Wang, J. Ma, C. Wang, and A. Kot, “Fault and intrusion tolerance of wireless sensor networks,” in *Proceedings of the 20th International Parallel and Distributed Processing Symposium*, 2006, p. 7.

[7] J. Deng, R. Han, and S. Mishra, “INSENS: Intrusion-tolerant routing in wireless sensor networks,” *Technical Report CU CS-939-02, Department of Computer Science, University of Colorado*, 2002.

[8] ——, “INSENS: Intrusion-tolerant routing for wireless sensor networks.” *Computer Communications*, vol. 29, no. 2, pp. 216–230, 2006.

[9] S. B. Lee and Y. H. Choi, “A secure alternate path routing in sensor networks,” *Computer Communications*, vol. 30, no. 1, pp. 153—165, December 2006.

[10] R. Xiuli and Y. Haibin, “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, pp. 174–178.

[11] L. Lamport, “Constructing digital signatures from one-way function,” *Technical Report SRI-CSL-98, SRI International*, 1979.

[12] S. Zhu, S. Setia, and S. Jajodia, “LEAP: Efficient security mechanisms for large-scale distributed sensor networks,” in *Proceedings of ACM CCS*, 2003.

[13] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, “A survey of key management schemes in wireless sensor networks,” *Comput. Commun.*, vol. 30, no. 11-12, pp. 2314–2341, 2007.

[14] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. E. Culler, and K. S. J. Pister, “System architecture directions for networked sensors,” in *Proceedings of Architectural Support for Programming Languages and Operating Systems*, 2000, pp. 93 – 104.

[15] L. Philip, L. Nelson, W. Matt, and C. David, “TOSSIM: Accurate and scalable simulation of entire tinyos applications,” in *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys 2003)*, 2003.

[16] B. L. Titzer, D. K. Lee, and J. Palsberg, “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. 67.

[17] C. Karlof, N. Sastry, and D. Wagner, “Tinysec: A link layer security architecture for wireless sensor networks,” in *Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SensSys 2004)*, November 2004.