SEIF: Secure and Efficient Intrusion-Fault Tolerant Routing Protocol for Wireless Sensor Networks

Abdelraouf Ouadjaout
USTHB, Algiers, Algeria
ouadjaout@gmail.com

Yacine Challal
UTC, Compiegne, France
ychallal@hds.utc.fr

Noureddine Lasla
INI, Algiers, Algeria
lnoureddine4@gmail.com

Miloud Bagaa
USTHB, Algiers, Algeria
bagmoul@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.

1 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*. 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 individual monitoring of sensors. Consequently, failures of nodes become *an inevitable phenomenon* which can reduce dramatically the overall network lifetime.

One solution for the network lifetime problem is to consider node failures as *a normal property of the network* and provide 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 [3, 8]. Among these schemes, building node disjoint paths has been considered as the most reliable one. Due to the absence of common sensors between disjoint paths, a link disconnection will cause at most *a single path to fail* for any sensor in the network. Hence, the impact of a failure is reduced to a minimum level.

In real deployments, security becomes another important issue [6]. In presence of malicious nodes, providing sensors with alternative paths is not sufficient to ensure a reliable system since many existing attacks can make the topology unusable. 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.

Existing intrusion-fault solutions suffer from a poor tradeoff between the scalability of the system and the level of tolerance offered by the produced topology. In this paper, we present two contributions to ensure this tradeoff:

- 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.

- 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

0-7695-3102-4/08 $25.00 © 2008 IEEE
DOI 10.1109/ARES.2008.128

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 rest of paper is organized as follows. Most representative intrusion-fault tolerance solutions are presented in section 2. We present our protocol SMRP in section 3. In section 4, we describe our secure and efficient intrusion-fault tolerant solution SEIF. Simulation results are detailed and analyzed in section 5. Finally, we summarize our work and draw conclusions in section 6.

2 Related Works

The first work on an intrusion-fault tolerant approach was the protocol INSENS [1]. It is based on a centralized link-state mechanism. At each round, every node sends to the sink the list of its neighbors with proofs of neighborhood. The sink can hence detect forged neighborhood information and will deliver to each node its corresponding routing table. 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 and requires a large amount of communication between sensors and the sink. A distributed version of INSENS, named EINSENS [2], was proposed to allow sensors making local decisions without referring to the sink. However, EINSENS can find only one path per sensor, but the authors *emulated* a multi-path routing by deploying several sinks and constructing a single route to each sink.

Lee et al. [7] 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. Moreover, SeRINS employs a *hybrid detection mechanism*. 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.

3 SMRP

In this section, we describe our protocol SMRP, an enhancement of node disjoint path construction for many-to-one communication paradigm.

3.1 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. *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 (TOSB), without any additional message requiring only one transmission per sensor [8]. The main idea of this type of routing is based on *tagging* any route message with the identification of the sinks 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.

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 cope with this limitation, H-SPREAD [8] proposed an extension to find more extra routes at cost of additional messages. 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.

3.2 Overview of our solution

In our solution, we have carefully redefined the nature of the alternative paths in order to preserve the constraint of using "one message per sensor"while increasing the number discoverable paths. In existing solutions, sensors reject automatically any message from an already discovered branch, in order to maintain the paths node-disjoint. To explore more routes without adding new messages, we have alleviated this constraint by allowing some particular nodes as intersection between paths.

Since the number of root nodes represents the number of discoverable branches, we can increase the number of alternative paths by assigning the tagging responsibility to the neighbors of root nodes (*i.e.* 2-hops neighbors of sink node). This way, we allow root nodes as intersection between routes and neighboring nodes of roots can become sub-roots and thereby construct their own sub-branches. A sensor will accept paths within the same branch only if they come from different sub-branches. 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.

3.3 Description

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

$$(r, parent, subBranch)$$

where r is the sequence number of the round, $parent$ is the ID of the sending node and $subBranch$ is 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.

3.3.1 Round initialization

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

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

3.3.2 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 subsequent RREQ messages with a new sub-branch tag, the sending node is selected as an alternative parent and the new route is added to the routing table.

3.3.3 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, if the sensor received a RREQ with an empty sub-branch, he 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.

4 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.

4.1 Problem definition

The unauthenticated tagging makes the branch-aware flooding mechanisms prone to different types of attacks. 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. To defend against these attacks, it is necessary to provide security countermeasures to authenticate the sub-branch origin, *i.e.* verify if the claimed sub-branch is really rooted at a trusted sub-root. Moreover, 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.

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.

As the tree construction can be repeated many times during the network life time, an attacker can himself initiate this action by spoofing the sinks 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 [6].

4.2 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*.A one-way hash chain (OHC) is a sequence of numbers $(K_i)_{0 \leq i \leq n}$ generated by a one-way function F as follows:

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

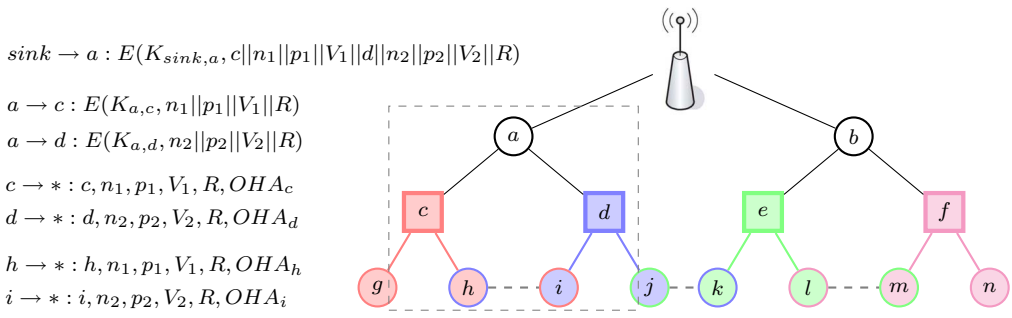

Figure 1. Illustrative execution of SEIF inside one branch, delimited with the dashed square.

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. When the sink starts a new round, it distributes to the sub-roots their respective valid 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:

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

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. However, this basic "one hop"authentication does not counter Hello Flooding attacks since it does not take into consideration the notion of reachable neighborhood. Since many key management protocols establish the key materials using *challenge-response mechanisms* [11], 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.

4.3 Detailed description

4.3.1 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 encrypted with BK_i. Neighboring nodes that are not newly deployed respond with the first unused value of their respective local chains.

4.3.2 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:

$$sink \rightarrow i: \quad E(K_{sink,i}, subRoot_1 || \ n_1 \ || \ p_1 \ || \ V_1 \ || \ ... \ || \ subRoot_m || \ n_m \ || \ p_m \ || V_m \ || \ R) \quad (1)$$

The structure of this message is as follows. $subRoot_k$ 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. 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 (1), 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:

$$i \rightarrow subRoot_k : E(K_{i,subRoot_k}, n_k \ || \ p_k \ || \ V_k \ || R) \quad (2)$$

After a sensor decrypts the message (2) 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.

4.3.3 Tree construction

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

$$i \rightarrow *: i, n, p, V, R, OHA_i \quad (3)$$

where n, p ,V and R represent the values received from the root node within the message (2). OHA_i is the first unused value of the local OHC for *one hop authentication*.

When a sensor j receives the message (3), it authenticates the sending node by verifying if $NV_{j,i} = F(OHA_i)$. 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 (1) and (2), 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 three conditions:

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

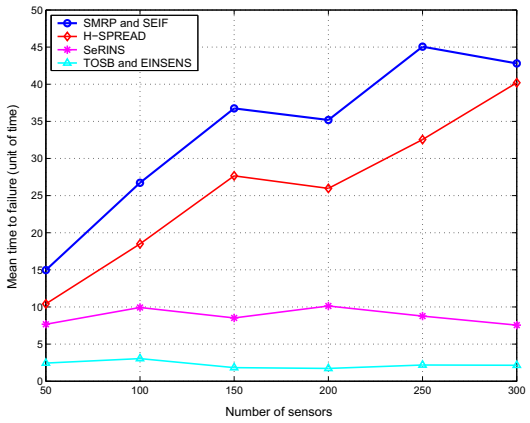

Figure 2. Mean Time To Failure (MTTF)

the sensor elects the sending node as an alternative parent and update $CV_{j,n}$ and $P_{j,n}$ with the received values. The parameter D defines the maximum number of iterations over function F to verify wether the received value belongs to the claimed chain. This will prevent an intruder from injecting large values of p to exhaust energy of its neighbors in useless calculations. After a random period of time after receiving the first RREQ in the round, the sensor node chooses randomly one main parent among the discovered alternative paths, and sends the message (3) using the sub-branch tag of the chosen main parent.

5 Simulations and Analysis

Using the TinyOS environment [4], we have implemented the following protocols: SMRP, SEIF, SeRINS, EINSENS, H-SPREAD and TOSB. 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 [9]. For a concise analysis of the energy consumption, we have used the Avrora tool [10]. For the family of secure protocols, we have used the TinySec library [5] for all cryptographic operations, such as encryption and hash functions. The number of sensors ranges from 50 to 300, while the average degree is fixed to 20.

5.1 Mean Time To Failure

The Mean Time To Failure (MTTF) 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. To evaluate this metric, we have simulated the protocols using TOSSIM to obtain the constructed

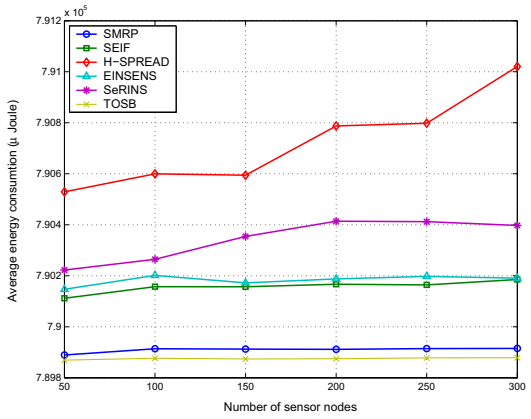

Figure 3. Average energy consumption

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* (such as months, years, ...*etc.*). For each failure, we check wether the graph is still connected. The summation of intervals between failures until graph disconnection gives the time to failure. To estimate the MTTF, we have executed 200 simulations for each scenario.

Fig. 2 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.

5.2 Energy consumption

Energy conservation is another compulsory goal in WSN architectures. One of the design goals of SMRP was to use only one message per node to conserve energy, while discovering more alternative paths. Fig. 3 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. For secure protocols, they have globally the same performance, with a slight advantage to our protocol SEIF.

6 Conclusion

In this paper, we have investigated the problems of fault tolerance and intrusion tolerance. 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 MTTF of our solution SEIF exceeds the MTTF of representative solutions in the literature.

References

[1] 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.

[2] J. Deng, R. Han, and S. Mishra. INSENS: Intrusion-tolerant routing for wireless sensor networks. *Computer Communications*, 29(2):216–230, 2006.

[3] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):11–25, 2001.

[4] 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*, pages 93 – 104, 2000.

[5] 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.

[6] 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*, 1(2–3):293–315, September 2003.

[7] S. B. Lee and Y. H. Choi. A secure alternate path routing in sensor networks. *Computer Communications*, 30(1):153—165, December 2006.

[8] 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*, 55(4):1320–1330, 2006.

[9] 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.

[10] 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)*, page 67, Piscataway, NJ, USA, 2005.

[11] 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.*, 30(11-12):2314–2341, 2007.