Towards a CDS-Based Intrusion Detection Deployment Scheme for Securing Industrial Wireless Sensor Networks

Lyes Bayou, Nora Cuppens, David Espes, Frédéric Cuppens

Towards a CDS-Based Intrusion Detection Deployment Scheme for Securing Industrial Wireless Sensor Networks

Lyes Bayou*, Nora Cuppens-Boulahia*, David Espes† and Frédéric Cuppens*
*Télécom Bretagne - LabSTICC,
2 Rue de la Châtaigneraie, Césson Sévigné, France,
Email: {lyes.bayou,nora.cuppens,frederic.cuppens}@telecom-bretagne.eu
†University of Western Brittany - LabSTICC, Brest,France
Email:david.espes@univ-brest.fr

***Abstract*—The use of wireless communication is a major trend in the so called Supervisory Control and Data Acquisition systems (SCADA). Consequently, Wireless Industrial Sensor Networks (WISN) were developed to meet real time and security requirements needed by SCADA systems. In term of security, WISN suffer from the same threats that those targeting classical WSN. Indeed, attackers mainly use wireless communication as a medium to launch these attacks. But as these networks are used to manage critical systems, consequences of such attacks can be more harmful. Therefore, additionally to the use of cryptographic and authentication mechanisms, Intrusion Detection Systems (IDS) are also used as a second line of defense. In this paper we propose an efficient IDS deployment scheme specially tailored to fit WISN characteristics. It builds a virtual wireless backbone that adds security purposes to the WISN. We also show that the proposed deployment scheme provides a good traffic monitoring capability with an acceptable number of monitoring nodes. It particularly allows detecting that a packet has been forged, deleted, modified or delayed during its transmission.**

I. Introduction

For several years, an important trend in industrial process management, has been the increasing use of wireless communication in the so called Supervisory Control and Data Acquisition systems (SCADA). SCADA systems are computer-based technology used for monitoring and managing industrial installations such as power plants, refineries, railways, etc. This is mainly due to the low-cost and the great flexibility provided by WSN that enhance significantly the sensing capability of SCADA systems. Therefore, Wireless Industrial Sensor Networks (WISN) are proposed to provide reliable, robust and secured communication in order to meet industrial requirements such as availability and real-time.

As a specific application of WSN, Wireless Industrial Sensor Networks (WISN) present the followings key characteristics [1] :

- They are expected to work reliably in industrial harsh environment: wide temperature range, vibrations, reflections due to metallic structures, etc.
- There is an online trusted party (the base station).
- There is no data aggregation. All sensing data are sent to the base station.
- Used protocols must be energy efficient. The battery life of sensors is expected to last several years.

In terms of security, WISN are subject to the same attacks as WSN and other wireless networks. Indeed, attackers mainly use wireless communication as a medium to launch their attacks. Moreover, as WISN manage sensitive installations and facilities, attacks against them can lead to harmful economical consequences or even can threaten human lives [2]. Therefore, several mechanisms were developed to enhance the security of these networks (Cryptography, Authentication, etc). But as these security solutions cannot prevent all attacks, especially in the case of node compromise attacks [3], Intrusion Detection Systems (IDS) are used as a second line of defense [4].

Nevertheless, we cannot directly apply intrusion detection techniques used in other wireless networks such as *ad hoc networks*, to supervise Wireless sensor networks without their adaptation. Indeed, there are three features that distinguish WSN and more specifically WISN from other wireless networks [5]:

1) processor, memory, energy and channel are limited resources;
2) WISN are usually not mobile;
3) the behavior of a WISN is highly predictable since it is composed of devices with few human interventions. So, communication and exchanged data respond to specific profiles in terms of quantity and frequency.

Depending on where the intrusion detection logic is implemented, these systems can be divided in two categories [6]: centralized and distributed systems. In centralized systems, an IDS agent connected to the WSN, mainly through the base station, analyzes information sent from wireless sensors in order to detect potential attacks. In decentralized systems, the detection logic is implemented directly into sensors called IDS-agents. These IDS-agents monitor the behavior of adjacent sensors. Hybrid systems consist of a central agent connected to the main station and IDS-agents deployed among sensors. By this way, both local and end-to-end communications are analyzed.

An important issue in such architectures is the deployment of IDS-agents. Indeed, the detection efficiency depends greatly on the collected data quality. Therefore, the localization of devices used to collect data must be well studied otherwise a part of communication will not be monitored.

We must note that as in *classical* WSN, clustering techniques are widely used for providing routing features and data aggregation, IDS-agents are generally implemented in the *cluster head* [5]. Nevertheless, this placement method is not efficient especially in WISN. Indeed, with this method we do not have the guaranty that all communications are monitored as only communications between sensors and cluster heads are checked by IDS-agents.

In this paper, we present a deployment scheme for the placement of the IDS-agent of a decentralized IDS in a Wireless Industrial Sensor Network. It presents the best trade-off between the number of used IDS-agents and the detection efficiency. We use the graph theory concept of *Dominating Set* to select nodes that will be substituted by super-nodes. Super-nodes have enhanced storage and processing capacities that allow them to act in the same way as normal sensors and also as detection agents. By this way, a virtual wireless backbone network providing intrusion detection capabilities will be created upon the WSN.

To validate the deployment scheme, communication in the context of WSN were modeled and then it was proven that this scheme fulfills the defined security requirements.

The rest of the paper is organized as follows. Section II presents some common attacks on WSN. In Section III, we discuss several intrusion detection techniques used in WSN. We show in Section IV how the deployment scheme have been ignored in almost all previous studies. Section V describes security requirements and the attacker model. Our deployment scheme is detailed in Section VI. A formal validation of this deployment is given in Section VII. The performances of the proposed scheme are presented in Section VIII. Finally, Section IX, presents the conclusion and future works.

II. WSN ATTACKS:

WSN can be subject to several kinds of attacks. These attacks can target important mechanisms such as [7]: routing protocol, data aggregation, voting, fair resource allocation, and misbehavior detection algorithms. We give below the description of some well-known attacks on WSN:

- Jamming attack: in this kind of attacks, a malicious node emits periodically or continuously on one or more channels. This will create interferences which will disturb transmissions of nearby nodes.
- Denial of Service (DoS) attack: it can be executed by flooding a node. A malicious node sends a great amount of packets to a node. The targeted node will be overwhelmed and will not be able to receive legitimate packets.
- Sinkhole and blackhole attacks: in this kind of attack a malicious node misleads routing algorithm by transmitting false information to the base station. As a result a part of the traffic will be redirected to the malicious node which can drop partially (wormhole) or totally (blackhole) packets.
- Selective forwarding attacks: a malicious node chooses selectively to drop some packets and to not forward them to their final destination.
- Wormhole attacks: in this kind of attacks a malicious node creates a virtual tunnel by capturing packets in one location and retransmits them in another location of the network. To do that, the malicious node must have a transmission range longer than other nodes or can require the help of another malicious node. As results, the malicious node can circumvent the routing protocol and lies on its location (number of hops from the base station).
- Hello Flood attacks: These attacks target some routing protocols, specially those using a certain kind of packet *hello packets*, to discover their immediate neighborhood. A malicious node with a large transmission range can flood a large part of the network with this kind of packets. Nodes receiving these packets, will assume that the malicious node is in their transmission range. As results, normal nodes can exhaust their battery life by trying to communicate with the malicious node.
- Sybil attacks: Sybil attacks were first described by Douceur in [8]. He shows that in the absence of a central identification authority that checks correspondence between entity and identity, a malicious entity can present multiple identities. This kind of attack can be used to target several types of protocols in WSN such as distributed storage, routing, data aggregation, voting, fair resources allocation and misbehavior detection algorithms [9].
- Forced delay attacks: a sensor node deliberately delays packets forwarding which can disturb significantly the installation functioning. This kind of attacks can have harmful consequences in WISN where processes are time sensitive.

III. INTRUSION DETECTION TECHNIQUES FOR INDUSTRIAL WIRELESS SENSOR NETWORKS

Several techniques are used in IDS to detect attacks such as watchdogs or local monitoring [10], spontaneous watchdogs [11], edge self-monitoring [12][13], etc. These techniques rely on the broadcast nature of wireless communication. Indeed, each node is able to overhear all packets sent by nodes in its neighborhood. Nevertheless, these techniques suffer from several drawbacks [14]. In local monitoring or watchdog, selected nodes are used for monitoring specific part of the wireless network. This technique requires that watchdog nodes overhear and store all exchanged packets in their neighborhood. Consequently, it is a very energy and computational resources consuming technique as watchdog nodes must be active continuously. In industrial process management, such technique is in practice not applicable since sensors have limited resources.

To reduce some of these drawbacks, spontaneous watchdog technique was proposed [11]. In this technique, all nodes implement a local agent that monitors information relative to the sensor node itself. Also, a global agent is activated randomly and acts as a watchdog. Thus, as global agent is not active continuously in each node, the added overload is lower in comparison to the previous technique. However, this technique does not ensure that all packets are overheard by a global agent which reduce significantly the IDS efficiency.

In edge self-monitoring technique [12], nodes are put in sleep or active mode in such a way that each transmission link is always monitored by k nodes (*k-self monitoring*). This technique ensures that all the traffic is monitored and node resources are not overused. The drawback is that monitoring nodes have partial information about monitored nodes. Indeed, the same node is not monitored each time by the same monitoring nodes. Consequently this technique is not efficient for intrusion detection.

IV. Related Work

After studying many intrusion detection systems specially designed for wireless sensor networks [5], we can conclude that the detection logic deployment issue is rarely mentioned. Indeed, although these studies use selected sensors for implementing totally or partially an intrusion detection logic, there is no indication how these sensors are selected. In [15], Da Sila et al. proposed one of the first intrusion detection systems for WSN. They designed a decentralized system in which a set of nodes is designated as *monitor* and is responsible of monitoring their neighbors looking for intruders. Nevertheless, it is not specified how these monitoring nodes are selected except that all nodes must be monitored.

In a more recent study, Roosta et al. proposed in [16] an intrusion detection system for wireless process control systems. The system consists of two components: a central IDS and multiple field IDS distributed among sensor nodes. These field IDS are deployed in *super-nodes* that passively monitor communications in their neighborhood. They periodically send collected data to the central IDS that will check their conformity with the security policy. Even if it is mentioned that the central IDS is implemented in the Network Manager (The base station in this kind of networks) there is no indication about the deployment of field IDS.

We also must note that in Wireless Sensor Networks, there are two others methods that are more or less similar to the IDS deployment which are *Base Station deployment* and *Network Clustering*.

In Base Station deployment studies, the aim is to find the optimal location of one or several base stations in order to ensure the radio coverage, reduce communication latency or increase the network lifetime [17]. The most important criteria here is the determination of the most suitable location for the base station that ensures reliable communications with nodes.

In the Network clustering studies, the aim is to organize the wireless network into a collection of small-size networks [18]. This is mainly used in routing protocol or in transmission bandwidth optimization. In both cases only nodes designated as *cluster heads* implement the routing table or have the ability to aggregate received data which reduces the redundancy and the network load. Clusters are built in such a way all nodes are located at *k*-hops on maximum from the *cluster head* or by sitting equal-size clusters to perform load balancing [18].

In IDS deployment issue, nodes selection criteria are different from those used in base station deployment or cluster heads selection. Indeed, IDS system must be deployed in such a way all exchanged packets are monitored and their conformity with the security policy is checked.

V. Security requirements and Attacker model

Industrial systems rely on processing the sensing data gathered from several kinds of sensors deployed throughout the facility. Therefore, to ensure the industrial process continuity in safe condition, it is important that data sent by these sensors are effectively received by the main station and in the appropriate time. In other words, we must be able to check that the right information arrives to the right destination at the right time without any modification, alteration or delay.

Consequently, we must be able to check the following security requirements:

- the packet source (non repudiation).
- the packet integrity (non modification).
- the packet delivery.
- the packet delivery time.

In this study, we assume that the aim of the attacker is to disturb the industrial process. This can be done either by dropping packets, injecting false packets or modifying packets. The attacker can also delay transmission of some important packets (alarms, sensing data, etc) to hide its malicious activity. Therefore, we consider in our study a Dolev-Yao like attacker [19] that can intercept, modify, forge or delay packets. For that it can only use its own credential (cryptographic keys) without any attack against used cryptographic mechanisms.

VI. The proposed CDS-Based deployment scheme

In this Section, we present our CDS-Based deployment scheme for securing Wireless Industrial Sensors Networks. We use for that the *Connected Dominating Set* (CDS), a well known concept in the Graph Theory.

The aim of this study is to propose an efficient scheme to select sensor nodes in which intrusion detection logic will be implemented. Thus, on the basis of an existing sensor network topology, selected nodes are substituted by enhanced nodes called *Super-Nodes*. These super-nodes will act as classical sensor nodes by fulfilling sensing tasks and will also implement intrusion detection capabilities. As a result, a virtual wireless backbone that provides security purposes will be created upon the network.

To achieve this goal, our approach relies on WISN communication characteristics:

1) Local communication: used by adjacent nodes, that acts as relay nodes, to forward packets from the sender to

their final destination. It is also used to exchange messages between adjacent nodes to maintain the network connectivity.
2) End-to-end communication: used to transmit sensing data or commands between nodes and the base station. Usually this kind of communication is encrypted.

In order to be efficient, an IDS must collect all exchanged packets in both local and end-to-end communication. Consequently, a two-level architecture is the appropriate choice. It consists in a central agent and several IDS-agents. The central agent is responsible for monitoring end-to-end communications and global coordination. It is implemented in the base station. IDS-agents are responsible for monitoring local communications of sensor nodes inside their neighborhood. They are implemented in selected sensor nodes.

Also, to be more efficient, the deployment scheme must fulfill two requirements:

1) each IDS-agent must be able to monitor its neighborhood without any cooperation with other IDS-agents. This requirement aims to decrease the overload added by the IDS and thus avoid to disturb the industrial process by adding a great amount of packets. Also, it allows a better detection effectiveness since an IDS-agent can by itself detect the attack.
2) each IDS-agent must be adjacent to at least another IDS-agent. This requirement aims to ensure that we have a secure communication channel between each IDS-agent and the central-IDS. Indeed, as IDS-agents are resilient nodes, we can always trust them for forwarding alarm packets especially in the case of nodes compromission.

A final but not less important point, is that the deployment scheme must be able to fulfill above requirements with an acceptable IDS-agents number. Indeed, the implementation of the detection logic in enhanced sensors capabilities must be cost efficient by reducing the number of these sensors.

According to the above requirements, our deployment scheme, that we call CDS-based deployment scheme, includes the three following steps: (i) *Connectivity Graph Construction*: A preparatory step in which the wireless sensor network is modeled by a graph called *Connectivity Graph*. (ii) *Connected Dominating Set Construction*: In this step, the connected dominating set is computed for selecting nodes that will be substituted by *IDS-agents*. (iii) *Uncovered Links Removal*: A final step that selects additional nodes for enforcing the monitoring coverage of some links.

A. Connectivity Graph Construction

In this step we model the wireless sensor network as a graph $G = (V, E)$, where V represents the set of all nodes in the network and E represents the set of all links in the network. Building the set V is straightforward as each node in the wireless networks is represented by a vertex. Modeling E is more sophisticated. Indeed, two nodes are *linked* if they can communicate which means that each node is in the transmission range of the other node.

Several models have been proposed to specify the transmission range. The most used model is the Unit disk graph (UDG). As illustrated in Fig. 1, two nodes are adjacent if and only if their Euclidean distance is at most 1 (or in general case less than a radius r). This model is idealistic as it assumes that the transmission range is uniform and omnidirectional and do not consider obstacles.

Other models try to be more realistic by considering waves propagation and path loss. Due to its simplicity and efficiency, the COST231 multi-wall model [20] is widely used in indoor environment [21]. In this model, the path loss in dB for environments with just one floor is given by L_{dB}, where the integer k_w represents the number of wall types, k_{wi} and L_{wi} represent respectively the number and the loss of the i^{th} wall type and $L_{0,dB}$ is the free space propagation to 1 meter.

$$L_{dB} = L_{0,dB} + 20 \log_{10} d + \sum_{i=1}^{k_{wi}} k_{wi} L_{wi} \quad (1)$$

The reader can refer to [21] and [22] to have more details about other models.

B. Connected Dominating Set (CDS)

In Graph Theory, a *Dominating Set* (DS) D of a graph G is a subset of nodes such that each node in G is adjacent to at least one node in D. A node in D is called a *dominator node*.

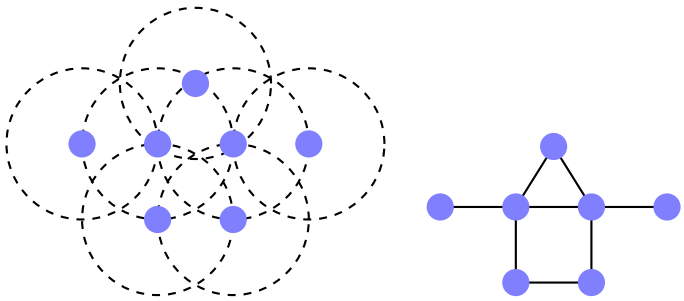
A *Connected Dominating Set* (CDS) is a dominating set in where each dominator node is adjacent to at least one other dominator.

In WSN, CDS have been used for creating a virtual backbone of the network (VBN). The VBN is mainly used as a spin for routing purposes [22]. Only nodes in the dominating set have routing features. Other nodes must send their messages to their closest dominator. Then, messages will be routed to the final destination throughout the VBN. The CDS can also be used for [22]:

- improving multicast/broadcast routing by restricting the forwarding of such messages to dominator nodes only,
- managing power consumption by making more nodes in a sleep mode,
- providing reliable and stable links.

Finding the *minimum* (connected) dominated set (M(C)DS) i.e., a CDS with the smallest size, is a NP-hard problem [22], [23]. Therefore many algorithms for constructing an approximate M(C)DS have been proposed. These algorithms can be divided into two categories: centralized algorithms and decentralized algorithms. Centralized algorithms are used under the assumption that the complete network topology is known. In decentralized algorithms, the dominator nodes are selected after a message exchange process between nodes.

The proposed deployment scheme relies on a centralized algorithm. Firstly, because this deployment scheme will be performed off-line on a topology-known WISN and also because centralized algorithms in general yield to a smaller CDS with a better performance ratio than decentralized algorithms [22].



(a) Wireless Sensor Network (b) Connectivity Graph

Fig. 1. Connectivity graph construction

Guha and Khuller propose in [23], a greedy centralized algorithm to construct a *Minimal* CDS and prove that it performs in a polynomial time. This algorithm builds a spanning tree rooted at the node that has a maximum degree. Each time a node is selected as a dominator, its neighbors are marked. The marked node with the maximum degree is selected as a dominator for the next step. The tree grows until all nodes are added to it. The non-leaf nodes in the tree form a CDS.

We propose a modified algorithm based on the Guha and Khuller algorithm. Indeed, instead of starting by the node with the maximum degree, we use the node representing the base station as a tree root. The pseudo-code of the proposed algorithm is given in Alg. 1.

Algorithm 1 CDS construction algorithm

▷ Black nodes are dominators
▷ Gray nodes are neighbors of dominators
▷ White nodes are not yet dominated
▷ Initially all nodes are white
Mark the node $root$ as black; ▷ Start from the base station
Mark $root$ neighbors as gray nodes;
while Exist a white node **do** ▷ repeat until all nodes are marked
    Select n the gray node with the maximum degree;
    Mark n as black;
    Mark node n neighbors as gray nodes;
end while

C. Uncovered Links Removal

To detect efficiently some attacks such as selective forwarding, message injecting or dropping, the IDS-agent should overhear packets received by a node and also those transmitted by that node. By construction, the IDS-agent CDS-based deployment scheme ensures that each packet transmitted by any node in the wireless network is overheard by at least one IDS-agent.

But as illustrated in Fig. 2, this does not guarantee that all packets received by that node are always overheard by at least one of IDS-agents monitoring the node receiving the packet. Indeed, IDS-agent in A cannot check that K actually retransmits packets received from F and the same holds for IDS-agent B for packets received by F from K.

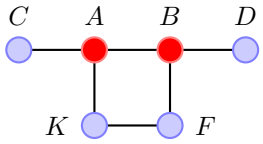


Fig. 2. Example of an uncovered link

Thus, after applying the CDS algorithm, we *can* have some *uncovered links* as the link $K - F$ in Fig. 2 .

Uncovered links are links that fulfill the two following conditions:

1) no one of their vertices is an IDS-agent (a dominator).
2) and also their vertices are monitored by different IDS-agents.

For monitoring these links, we implement an algorithm for their detection as illustrated in Alg. 2. This algorithm starts by building the list of uncovered links. Then, it marks as dominator the node that is part of the maximum number of uncovered links. After updating the uncovered links list, it repeats previous actions until all uncovered links are monitored.

Algorithm 2 Uncovered links detection and monitoring algorithm

Build L the list of uncovered links;
while L is not empty **do** ▷ repeat until all uncovered links are monitored
    Select n the node part of the maximum number of uncovered links;
    Mark n as black;
    Update L;
end while

VII. Deployment scheme formal validation :

To validate the proposed deployment scheme, we first define communication properties of wireless communication. Then, we specify both attacker capabilities and security requirements. Finally, we prove that the defined security requirements are completely fulfilled by the deployment scheme properties.

A. Notation :

Let us assume the following:

- V and D represent respectively, the set of nodes and the set of IDS-agents with $D \subset V$.
- M represents the set of all exchanged packets.
- $sendPacket(n_1, n_2, n_3, n_4, m, t)$ means that node n_1 sends to node n_2 the packet m originated from the node n_3 and destined to n_4 at time t.
- $receivePacket(n_1, n_2, n_3, n_4, m, t)$ means that the node n_1 receives from the node n_2 the packet m originated from the node n_3 and destined to the node n_4 at the time t.
- $neighbors(n_1, n_2)$ means that the node n_1 is the neighbor of the node n_2.
- $equivalent(m, m')$ means that packet m' is the forwarded version of the packet m and only fields in the header have been changed according to the used communication protocol.
- ϵ represents the propagation delay of a packet.
- δ represents the maximal time a packet must be forwarded within.
- δ' (with $\delta << \delta'$) represents the maximal time a packet is considered as deleted if it has not been forwarded.

We must note that in the predicates $sendPacket$ and $receivePacket$ the final destination of the packet m is the node n_4 and that the node n_2 is used as relay to forward this packet to its final destination.

B. Properties definitions:

- WISN properties:
  1) Medium broadcast property: as regards to the broadcast nature of wireless medium, a packet m sent by a node n_1 is received by all its neighbors.

$$\begin{aligned} &\forall n, n_1, n_2, n_3, n_4 \in V, \\ &sendPacket(n_1, n_2, n_3, n_4, m, t) \wedge \\ &neighbors(n_1, n) \\ &\Rightarrow receivePacket(n, n_1, n_3, n_4, m, t + \epsilon) \end{aligned}$$

  2) Channel symmetry property: If node n_1 is a neighbor of node n_2, node n_2 is also a neighbor of node n_1.

$$\begin{aligned} &\forall n_1, n_2 \in V, \\ &neighbors(n_1, n_2) \Leftrightarrow neighbors(n_2, n_1) \end{aligned}$$

  3) Multi-hop property: If node n_1 receives a unicast packet m originated from the node n, so either n_1 and n are neighbors, or there is a node n_2 neighbor of node n_1 that has forwarded this packet.

$$\begin{aligned} &\forall n_1, n_2, n, n' \in V, m \in M, t \in T, \\ &receivePacket(n_1, n_2, n, n', m, t) \\ &\Rightarrow (sendPacket(n_2, n_1, n, n', m, t - \epsilon) \wedge \\ &neighbors(n_2, n_1)) \end{aligned}$$

- Attacker properties:
  1) Forging a fake packet: in this attack, a malicious node n_1 pretends retransmitting to n_3 a packet m received from the node n_2.

$$\begin{aligned} &\forall n_1, n_3, n, n' \in V, n_1 \neq n, m \in M, t \in T, \\ &forgePacket(n_1, n_3, n, n', m, t) \\ &\Rightarrow sendPacket(n_1, n_3, n, n', m, t) \\ &\wedge \neg(\exists m' \in M, \exists n_2 \in N, \exists t' \in T, t' < t, \\ &receivePacket(n_1, n_2, n, n', m', t') \\ &\wedge equivalent(m, m') \\ &\wedge neighbors(n_1, n_2) \wedge neighbors(n_1, n_3)) \end{aligned}$$

  2) Deleting a packet: in this attack, a malicious node n_1 does not forward to the next node n_2 a received packet m destined to the node n' within the defined time δ'.

$$\begin{aligned} &\forall n_1, n_2, n_3, n, n' \in V, n_1 \neq n', m \in M, t \in T, \\ &deletePacket(n_1, n_2, n, n', m, t + \delta) \Rightarrow \\ &receivePacket(n_1, n_3, n, n', m, t) \\ &\wedge \neg(\exists m' \in M, \exists t' \in T, t + \epsilon < t' < t + \delta', \\ &sendPacket(n_1, n_2, n, n', m', t') \\ &\wedge equivalent(m, m') \wedge neighbors(n_1, n_2)) \end{aligned}$$

  3) Modifying a packet: in this attack, a malicious node n_1 forwards to the next node n_2 a packet m' which is a modified version of a received packet m.

$$\begin{aligned} &\forall n_1, n_2, n_3, n, n' \in V, m \in M, t \in T, \\ &modifyPacket(n_1, n_2, n, n', m, t) \Rightarrow \\ &receivePacket(n_1, n_3, n, n', m, t') \\ &\wedge (\exists m' \in M, \exists t' \in T, \\ &receivePacket(n_2, n_1, n, n', m', t)) \\ &\wedge \neg equivalent(m, m') \wedge neighbors(n_1, n_2) \end{aligned}$$

  4) Delaying a packet: in this attack, a malicious node n_1 forwards to the next node n_2 the received packet m after the defined time δ.

$$\begin{aligned} &\forall n_1, n_2, n_3, n, n' \in V, m \in M, t \in T, \\ &delayPacket(n_1, n_2, n, n', m, t) \Rightarrow \\ &\exists m' \in M, \exists t' \in T, t' + \delta < t < t' + \delta', \\ &receivePacket(n_1, n_3, n, n', m, t') \\ &\wedge sendPacket(n_1, n_2, n, n', m', t) \\ &\wedge equivalent(m, m') \wedge neighbors(n_2, n_3) \end{aligned}$$

- WISN Security requirements:

1) Traffic monitoring property: In order to gather all exchanged traffic, the IDS system i.e., all IDS nodes, must receive all sent messages.

$$\begin{aligned}\forall n_1, n_2, n, n' \in V, \forall m \in M, \forall t \in T,\\ \exists d \in D, sent(n_1, n_2, n, n', m, t) \Rightarrow\\ receivePacket(d, n_1, n, n', m, t + \epsilon) \wedge\\ neighbors(d, n_1)\end{aligned}$$

2) Forged packet Detection property: an IDS-agent d receives the packet m sent by n_1 to n_2 without receiving the equivalent packet m' sent to n_1 by n_3.

$$\begin{aligned}\forall n_1, n_2, n_3, n, n' \in V, n_1 \neq n, \forall m \in M, \forall t \in T,\\ \exists d \in D,\\ detectForgedPacket(d, n_1, n_2, n, n', m, t)\\ \Rightarrow \neg(\exists m' \in M, \exists t' \in T, t' < t,\\ receivePacket(d, n_3, n, n', m', t')\\ \wedge receivePacket(d, n_1, n, n', m, t)\\ \wedge equivalent(m, m')\\ \wedge neighbors(d, n_1) \wedge neighbors(d, n_3))\end{aligned}$$

3) Deleted packet Detection property: an IDS-agent d receives the packet m sent by n_3 to n_1 but does not receive the equivalent packet m' forwarded by n_1 to n_2 within the defined time δ'.

$$\begin{aligned}\forall n_1, n_2, n_3, n, n' \in V, n_1 \neq n', \forall m \in M,\\ \forall t, t' \in T, t < t' < t + \delta', \exists d \in D,\\ detectDeletePacket(d, n_1, n_2, n, n', m, t + \delta')\\ \Rightarrow \exists m' \in M, receivePacket(d, n_3, n, n', m', t)\\ \wedge \neg receivePacket(d, n_1, n, n', m, t')\\ \wedge equivalent(m, m') \wedge neighbors(n_1, n_3)\end{aligned}$$

4) Modified packet Detection property: an IDS-agent d detects that the packet m sent by n_3 to the node n_1 and the packet m' forwarded by n_1 to n_2 are not equivalent.

$$\begin{aligned}\forall n_1, n_2, n_3, n, n' \in N, \forall m \in M, \forall t, t' \in T,\\ \exists d \in D,\\ detectModifyPacket(d, n_1, n_2, n, n', m, t)\\ \exists m' \in M, \Rightarrow receivePacket(d, n_3, n, n', m, t) \wedge\\ receivePacket(d, n_1, n, n', m', t')\\ \wedge \neg equivalent(m, m')\\ \wedge neighbors(n_1, n_3) \wedge neighbors(n_1, n_2)\end{aligned}$$

5) Delayed packet Detection property: an IDS-agent d that receives the packet m sent from the node n_3 to the node n_1, also receives the packets m' that n_1 forwarded to the next node n_2 after the defined time δ.

$$\begin{aligned}\forall n_1, n_2, n_3, n, n' \in V, \forall m \in M,\\ \forall t \in T, \exists d \in D,\\ detectDelayPacket(d, n_1, n_2, n, n', m, t')\\ \Rightarrow \exists m' \in M, \exists t' \in T, t + \delta < t' < t + \delta',\\ receivePacket(d, n_3, n, n', m, t)\\ \wedge receivePacket(d, n_1, n, n', m', t')\\ \wedge equivalent(m, m')\end{aligned}$$

- IDS Deployment properties:
  1) CDS property: a node is either an IDS or has at least one IDS as a neighbor.

$$\forall n \in V, \exists d \in D, neighbors(n, d)$$

  As each node has at least an IDS as a neighbor (CDS property) and each packet sent by a node is received by all its neighbors (the medium broadcast property), each sent packet is received by at least one IDS.

$$\begin{aligned}\forall n_1, n_2, n \in V, \forall m \in M, \forall t \in T, \exists d \in D,\\ sendPacket(n_1, n_2, n_3, n, m, t)\\ \Rightarrow receivePacket(d, n_1, n_3, n, m, t + \epsilon) \wedge\\ neighbors(n_1, d)\end{aligned}$$

  2) Uncovered link monitoring property: this property guarantee that there is always and IDS-agent d neighbor of two neighbor nodes n_1 and n_2.

$$\begin{aligned}\forall n_1, n_2 \in V, \exists d \in D,\\ neighbors(n_1, n_2)\\ \Rightarrow neighbors(d, n_1) \wedge neighbors(d, n_2)\end{aligned}$$

C. Security requirements guarantee proof:

Theorem 1: Deployment scheme properties guaranty WISN security requirements validation:

$$\begin{aligned}\forall n_1, n_2, n_3, n, n' \in V, \forall m \in M, \forall t \in T\\ neighbors(n_1, n_2) \wedge neighbors(n_1, n_3) \Rightarrow\\ \exists d \in D, detectForgePacket(d, n_1, n_2, n, n', m, t)\\ \wedge detectDeletePacket(d, n_1, n_2, n, n', m, t)\\ \wedge detectModifyPacket(d, n_1, n_2, n, n', m, t)\\ \wedge detectDelayPacket(d, n_1, n_2, n, n', m, t)\end{aligned}$$

Indeed, according to *the Uncovered link monitoring property*, if n_1 sends a packet m to its neighbor n_2 there is always an IDS-agent d, neighbors of n_1 and n_2 that receives the sent packet. Also, according to the medium broadcast property, the IDS-agent d receives all packets sent by n_2 and particularly the packet m' i.e., the forwarded version of the packet m. Consequently, the IDS-agent d can always compare packets m

and m' and checks if ever a packet have been forged, deleted, modified or delayed.

VIII. Performances Evaluation:

A. Dominating Nodes Ratio:

For evaluating the proposed deployment scheme performances, we conduct series of test on simulated wireless sensor networks. For this purpose, we use NS3 to deploy randomly n nodes in a rectangular field. Then, we vary the radius r, representing the transmission range of nodes. By that way, we get networks with different topology density (TD) that is the average node degree.

The dominating node ratio indicates the number of dominating nodes in a WSN compared to the total number of its nodes. For assessing the impact of the topology density on this ratio, we generate for each graph size, 50 random graphs with different topology density. Then, we measure for each generated graph, the dominating node ratio. In order to get accurate results, we measure the dominating node ratio of several generated random graphs with the same size and topology density. Then, we take the average of these measures that we illustrate in Fig. 3.

As intuitively expected, the dominating node ratio decreases, for all graph sizes, with the increase of the topology density. This ratio is about 30% with a TD equal to 7-8 and reaches 20-25% with a TD above 10-12.

We should note, that according to the best practices in WISN deployment [24], 25% of sensors should have a direct connection to the main station; each node should have at least 3 direct neighbors; and each node should not be 4 hops away from the main station.

Table I illustrates the dominating node ratio result for Alg. 1 and Alg. 2. We can see clearly that Alg. 2 does not add a great number of IDS-agents. Indeed, the maximum ratio of added IDS-agents is about 3.5 %. Thus, detecting and removing uncovered links strengthen the efficiency of the solution without increasing significantly the number of IDS-agents.

TABLE I
Dominating node ratio by algorithm.

n	Alg.1		Alg.2		Total Ratio (%)
	Result	Ratio (%)	Result	Ratio (%)	
50	24.23	48.46	0.15	0.30	48.76
100	42.38	42.38	1.15	1.15	43.53
200	72.15	35.92	4.69	2.23	38.15
300	102.33	33.75	9.08	3.00	36.75
400	109.07	27.00	13.43	3.57	30.57
500	133.69	26.46	16.84	3.30	29.76
600	149.76	24.46	21.39	3.69	28.15
800	166.78	20.42	27.29	3.50	23.92

B. Dominating nodes selection execution Time

As the CDS-Based deployment scheme is executed once and offline, it does not require a fast execution. Nevertheless, the average time taken by the execution of both Alg. 1 and Alg.2, illustrated in Fig. 4, shows that it takes very acceptable values. These performances are mainly due to the Alg. 1 that is executed in a polynomial time.

C. Traffic monitoring efficiency

Table II illustrates the number of nodes monitored by the same dominator node. We can see that the average number of nodes dominated by the same dominator node is always bigger than the Topology density, i.e., the average node degree. This is due to the Alg. 1 and Alg. 2 that choose dominator nodes with higher degrees.

TABLE II
Number of monitored nodes by a dominator.

n	TD	Number of dominated nodes		
		Min	Max	Avg
50	2.09	2.01	11.63	4.80
100	3.69	2.20	20.76	6.74
200	7.07	2.83	38.67	10.80
300	6.81	2.41	43.94	9.90
400	7.10	2.32	49.78	9.86
500	8.50	2.48	60.08	11.79
600	8.11	2.20	64.13	10.90
800	8.87	2.17	75.65	11.49

We can also see that a dominator node monitors at least 2 nodes which indicates that leaf nodes are never selected as dominating nodes.

In another hand, the maximum number of monitored nodes by the same dominator may seem higher particularly for networks with high density.

We must note in these cases that generally communication protocols for WISN use techniques such as Time Division Multiple Acces (TDMA) to manage transmission and avoid collisions. In these techniques, the bandwidth is divided into several channels (Typically 15 or 16) and only one transmission is allowed in the same channel at the same time. Consequently, the maximum number of communication that a dominator node has to monitor is equal to the number of transmission channels (15 or 16).

In Table III, we report the number of dominator nodes that monitors the same node. As expected, all nodes are at least monitored by one dominator. We can also see that on average, a node can be monitored by 2, 3 or more dominator nodes. Thus, this increases the detection efficiency as a node is monitored by several dominator nodes.

IX. Conclusion and Future Works

In this paper, we present an efficient IDS-agent deployment scheme for wireless sensor networks. This deployment scheme

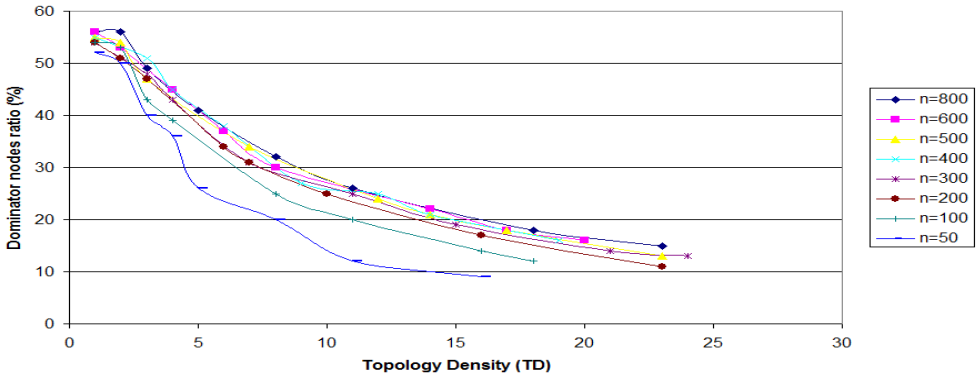


Fig. 3. Dominating nodes ratio compared to the Topology Density

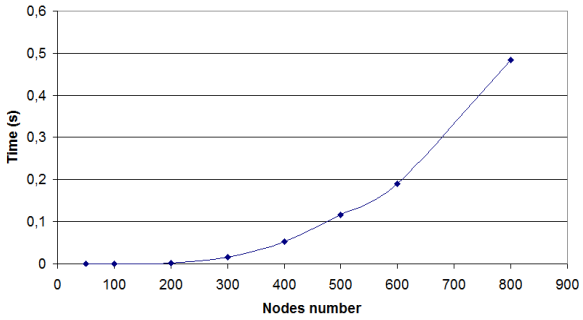


Fig. 4. Dominating nodes selection time

TABLE III
NUMBER OF DOMINATORS MONITORING A NODE.

n	TD	Number of dominated nodes		
		Min	Max	Avg
50	2.09	1	4.81	2.12
100	3.69	1	5.56	2.42
200	7.07	1	6.30	2.82
300	6.81	1	6.63	2.80
400	7.10	1	7.12	2.90
500	8.50	1	7.20	3.05
600	8.11	1	7.24	2.99
800	8.87	1	7.50	3.06

can be used either in decentralized, clustered or hierarchical architectures. It creates a virtual backbone that adds security purposes to an existing sensor network. To the best of our knowledge, it is the only complete deployment scheme implemented for security purposes. It presents good results both in terms of selected IDS-agent and execution time.

We must also note that the proposed deployment scheme fulfills totally WISN requirements especially in terms of communication specifications.

This work can be improved by different ways. For example, as several nodes are monitored by a great number of dominators, we can try to eliminate redundant dominators. Also, we can adapt the deployment scheme to be used in heterogeneous networks in which devices do not have the same capabilities in terms of transmission range, storage and computational resources. In this case, we can use weighted graphs to select nodes with higher capabilities firstly.

REFERENCES

[1] J. R. Moyne and D. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 29–47, Jan 2007.

[2] Y.-L. Huang, A. A. Cárdenas, S. Amin, Z.-S. Lin, H.-Y. Tsai, and S. Sastry, "Understanding the physical and economic consequences of attacks on control systems," *International Journal of Critical Infrastructure Protection*, vol. 2, no. 3, pp. 73 – 83, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1874548209000213

[3] L. Bayou, D. Espes, N. Cuppens-Boulahia, and F. Cuppens, "Security issue of wirelesshart based SCADA systems," in *Risks and Security of Internet and Systems - 10th International Conference, CRiSIS 2015, Mytilene, Lesbos Island, Greece, July 20-22, 2015, Revised Selected Papers*, ser. Lecture Notes in Computer Science, C. Lambrinoudakis and A. Gabillon, Eds., vol. 9572. Springer, 2015, pp. 225–241. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-31811-0_14

[4] I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks," in *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005), IEEE International Conference on*, vol. 3, Aug 2005, pp. 253–259 Vol. 3.

[5] R. Mitchell and I. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, vol. 42, pp. 1–23, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2014.01.012

[6] L. Coppolino, S. D'Antonio, L. Romano, and G. Spagnuolo, "An intrusion detection system for critical information infrastructures using wireless sensor network technologies," in *Critical Infrastructure (CRIS), 2010 5th International Conference on*, Sept 2010, pp. 1–8.

[7] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003. [Online]. Available: http://dx.doi.org/10.1016/S1570-8705(03)00008-8

[8] J. R. Douceur, "The sybil attack," in *Peer-to-Peer Systems, First International Workshop, IPTPS 2002, Cambridge, USA*, ser. Lecture Notes in Computer Science, P. Druschel, M. F. Kaashoek, and A. I. T. Rowstron, Eds., vol. 2429. Springer, 2002, pp. 251–260. [Online]. Available: http://dx.doi.org/10.1007/3-540-45748-8_24

[9] J. Newsome, E. Shi, D. X. Song, and A. Perrig, "The sybil attack in sensor networks: analysis & defenses," in *Proceedings of the Third International Symposium on Information Processing in Sensor Networks, IPSN, USA*, K. Ramchandran, J. Sztipanovits, J. C. Hou, and T. N. Pappas, Eds. ACM, 2004, pp. 259–268. [Online]. Available: http://doi.acm.org/10.1145/984622.984660

[10] I. Khalil, S. Bagchi, and N. B. Shroff, "Liteworp: Detection and isolation of the wormhole attack in static multihop wireless networks," *Computer Networks*, vol. 51, no. 13, pp. 3750 – 3772, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128607001004

[11] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, vol. 1, Jan 2006, pp. 640–644.

[12] D. Dong, X. Liao, Y. Liu, C. Shen, and X. Wang, "Edge self-monitoring for wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 3, pp. 514–527, March 2011.

[13] B. Neggazi, M. Haddad, V. Turau, and H. Kheddouci, "A self-stabilizing algorithm for edge monitoring problem," in *Stabilization, Safety, and Security of Distributed Systems - 16th International Symposium, SSS 2014, Paderborn, Germany, September 28 - October 1, 2014. Proceedings*, ser. Lecture Notes in Computer Science, P. Felber and V. K. Garg, Eds., vol. 8756. Springer, 2014, pp. 93–105. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-11764-5_7

[14] A. Abduvaliyev, A. K. Pathan, J. Zhou, R. Roman, and W. Wong, "On the vital areas of intrusion detection systems in wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 15, no. 3, pp. 1223–1237, 2013. [Online]. Available: http://dx.doi.org/10.1109/SURV.2012.121912.00006

[15] A. P. R. da Silva, M. H. T. Martins, B. P. S. Rocha, A. A. F. Loureiro, L. B. Ruiz, and H. C. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Q2SWinet'05 - Proceedings of the First ACM Workshop on Q2S and Security for Wireless and Mobile Networks, Montreal, Quebec, Canada, October 13, 2005*, A. Boukerche and R. B. de Araujo, Eds. ACM, 2005, pp. 16–23. [Online]. Available: http://doi.acm.org/10.1145/1089761.1089765

[16] T. Roosta, D. K. Nilsson, U. Lindqvist, and A. Valdes, "An intrusion detection system for wireless process control systems," in *IEEE 5th International Conference on Mobile Adhoc and Sensor Systems, MASS,USA*. IEEE, 2008, pp. 866–872. [Online]. Available: http://dx.doi.org/10.1109/MAHSS.2008.4660125

[17] K. Akkaya, M. Younis, and W. Youssef, "Positioning of base stations in wireless sensor networks," *IEEE Communications Magazine*, vol. 45, no. 4, pp. 96–102, April 2007.

[18] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14 - 15, pp. 2826 – 2841, 2007, network Coverage and Routing Schemes for Wireless Sensor Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366407002162

[19] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–207, 1983. [Online]. Available: http://dx.doi.org/10.1109/TIT.1983.1056650

[20] COST 231, "Digital mobile radio towards future generations systems," http://www.lx.it.pt/cost231/final_report.htm, European Commission, Final Report, 1999.

[21] C. B. Andrade and R. P. F. Hoefel, "IEEE 802.11 WLANs: A comparison on indoor coverage models," in *Proceedings of the 23rd Canadian Conference on Electrical and Computer Engineering, CCECE 2010, Calgary, Alberta, Canada, 2-5 May, 2010*. IEEE, 2010, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/CCECE.2010.5575205

[22] J. Yu, N. Wang, G. Wang, and D. Yu, "Connected dominating sets in wireless ad hoc and sensor networks - A comprehensive survey," *Computer Communications*, vol. 36, no. 2, pp. 121–134, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2012.10.005

[23] S. Guha and S. Khuller, "Approximation algorithms for connected dominating sets," *Algorithmica*, vol. 20, no. 4, pp. 374–387, Apr. 1998. [Online]. Available: http://dx.doi.org/10.1007/PL00009201

[24] HART Communication Foundation, "WirelessHART," http://www.hartcom.org.