International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 14, No.1 (Aug-2023)

http://dx.doi.org/10.12785/ijcds/140146

An Enhanced Dissection of Attacks in Wireless Sensor Networks

Benjamin Savoudsou[1], Franklin Tchakounté[1,2,3,6], Blaise Omer Yenké[4], Tiguiane Yélémou[5] and Marcellin Atemkeng[6]

[1]*Department of Mathematics and Computer Science, Faculty of Science, University of Ngaoundéré, Ngaoundéré, Cameroon*
[2]*SFTI, National School of Chemical Engineering and Mineral Industries, University of Ngaoundéré, Ngaoundéré, Cameroon*
[3]*Cybersecurity with Computational and Artificial Intelligence Research Group, Faculty of Science, University of Ngaoundéré, Ngaoundéré, Cameroon*
[4]*Department of Computer Engineering, University Institute of Technology, University of Ngaoundéré, Ngaoundéré, Cameroon*
[5]*Higher School of Computer Engineering, University of Nazi Boni, Burkina Faso*
[6]*Department of Mathematics, Rhodes University, Grahamstown 6140, South Africa*

Received 14 Jun. 2022, Revised 08 May 2023, Accepted 12 May 2023, Published 01 Aug. 2023

Abstract: Developing reliable solutions against attacks in wireless sensor networks (WSNs) requires dissecting the attacks to understand activities and strategies exploited by the attackers. Authors who proposed this comprehension consider active attacks and attacks occurring on specific layers but lacks to investigate relevant aspects such as the physical and logical components involved in the attack, routing protocols exploited and the position of attacker. We propose in this paper, a more complete Unified Modelling Language (UML) characterization of attacks which represent static and dynamic aspects of the attacker activities. Sixteen popular attacks have been studied and classified based on similarities and differences. For each attack, it is able to identify data that are exchanged, layers which are traversed, sequence of activities involved and components which are exploited within each attack. As a road-map to design countermeasures, resemblances and divergences are identified and discussed. A theoretical comparison with similar works has been made to show its complementarity.

Keywords: WSN, UML, Attack, Characterization

1. Introduction

Wireless Sensor Networks (WSN) is an infrastructure which enables the federation of wireless sensors and actuators to monitor and record physical or environmental events transferable to the final user. This technology is exploited in diverse applications such as agriculture, smart cities, automotive, connected industry, environmental monitoring, healthcare and breeding. Due to the affordability of sensors [1] and its role in sustainable economic growth, its demand increases. According to [2], its market value is forecasted at USD 203.94 billion by 2028 with an expected annual growth rate of 16.79% from 2021 to 2028.

As shown in Figure 1, WSN includes a group of spatially distributed tiny sensors and routing nodes which collect, process data and wirelessly route them to one or many base stations [3]. The base station is the final destination where the routing ends. It is directly (via cable) or wirelessly (via Internet, satellite or any other wireless links) connected to the user. Such networks have some properties based on node characteristics: (i) Nodes can be homogenous or heterogenous, (ii) sensors can remain immobile from their deployment or not, (iii) according to the structure of the network, sensors may play the same role or be dedicated to another function (iv) and sensors may send information directly to the base station or to relays.

WSN can be deployed following different architectures such as two-tier, three-tier, multi-tier and layered [4]. Relying on Open Systems Interconnection (OSI) model, the layered architecture is the popular one and thus it is the focus in this research. Its architecture in five layers is illustrated in Figure 2. The top layer i.e. physical layer, is used to transfer data bits on the communication channel. The second layer, namely data link layer, is exploited to guarantee reliable connectivity among nodes. The third layer, namely network layer, is used to route information to the destination. The fourth layer, namely, transport layer

E-mail address: savoudsoubenjamin18@gmail.com, f.tchakounte@cycomai.com, byenke@yahoo.com, m.atemkeng@ru.ac.za

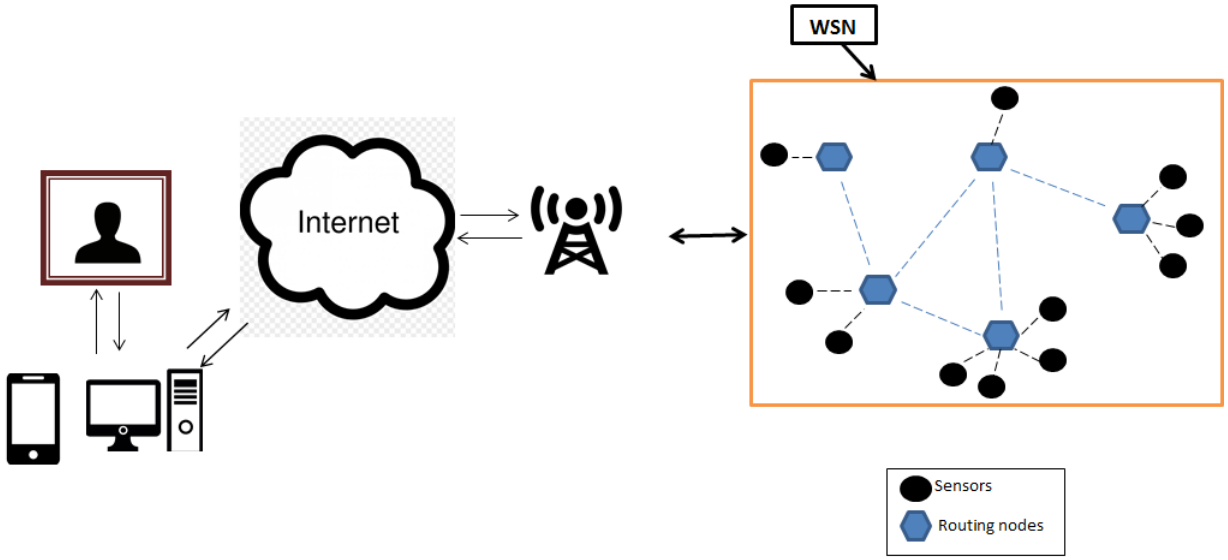

Figure 1. WSN Architecture

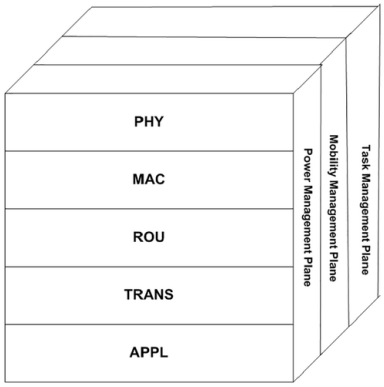

Figure 2. Layers in WSN [4]

is responsible to control congestion and ensure reliable delivery of messages to the base station [5]. The fifth layer, namely application layer, is responsible to let the user designing applications to exploit the WSN. The layers are supported by three packages such as power management, mobility management and task management respectively in charge of managing the power consumption of nodes, their mobility and locations and allocation of tasks [6], [7].

As much as WSN technology is finely structured to assist humans, it is subject to power-consumption and high-computing constraints [8]. Its integration in the IoT is advantageous because individuals can be connected independently from time, place, and service provided [9].

However, these capabilities interest attackers who develop active and passive attacks to eavesdrop, and compromise sensors and their communications. During an active attack, the functions and operations within the network are altered whereas in passive attacks, the adversary is sniffing and capturing without leaving any traces. Security in WSNs is more critical if they are operated for sensitive tasks directly exposing people's lives. For example, within the military and health frameworks, a simple compromise will result in loss of life resulting from false recommendations. The attacker proceeds differently in both active and passive attacks. Its mode of operation exploits specific elements and components in the different layers. Sometimes it re-uses some features of similar attacks depending on the objective to be achieved. Due to that, dynamic attacks quickly arise.

From literature, there are quite huge amount of countermeasures including cryptography mechanisms to hide data in communications [10], [11], intrusion detection techniques enhanced nowadays with artificial intelligence [12], [13]. Although they are interestingly dedicated to mitigate WSN infiltrations, they may be ineffective and inappropriate in different situations. They are ineffective because they are

not exhaustive in terms of considering fine-grained aspects exploited within the attack. For example, components target in different layers provide orientations to reinforce patching. They are inappropriate because cyber professionals try to apply techniques which fit in one category in another category. For example, detection in active attacks involves real play of communications which is not the case in passive attacks. In light of these reasons, depth dissection of attacks should be provided to support more technical ones.

Few works are proposed in the literature in this direction. Characterization of attacks are described and formalized with UML representation in [14]–[17]. Nevertheless, none of them provide simultaneously rudiments to reinforce effectiveness and appropriateness in technical solutions. Indeed, they do not combine in the same the dimensions of layer, type of attacks and type of infiltration.

This paper contributes with a dissection of attacker strategies in terms of (1) targeted layers with physical and logical features which are consciously manipulated by attackers, (2) aspects related to both internal and external perpetrations, and (3) typologies of attack semantics. The foremost added value is the fusion of such parameters during the investigation of 16 active and passive attacks.

During the research, attacks are classified by layers. For the sake of clearness, each one is described and designed using Unified Modelling Language (UML) sequence diagrams in which fine aspects are illustrated. Then, the components are discussed to demonstrate its exploitation in the attack. Attacks that are common to the others are also outlined. A map view of similarities is provided so that anyone can use as a support in this domain.

The remainder of this paper is organized as follows. Similar works with a focus on characterisation proposals are discussed in section 2. In section 3, the gap to fill is given with justifications. In section 4, dissection of attacks is provided. In section 5, a theoretical comparison with similar studies based on some criteria is provided. At the end, the paper is concluded and future paths are suggested.

2. Related works

In this section, existing literature orientations to counter attacks are described.

A. Cryptography-based mechanisms

In this category, different approaches are used to counter attacks within WSNs such as [18] and [19]: data partitioning, cryptosystems, key management, authentication, trust models. These approaches require the use of resources, although rare in WSNs.

B. Detection and prevention solutions

Authors here develop solutions dealing with prevention and detection of malicious infiltration. To achieve this objective, they rely on artificial intelligence [20], [21] to provide robust IDS [22]. In this category, authors need to identify ways and means by which attackers infiltrate the vulnerable nodes. For that, a concise dissection of attacks is required.

C. Characterization of attacks

The design of security mechanisms as well as more secure intrusion detection systems requires a good knowledge of sensor networks as well as a perfect understanding of attack scenarios, which is why several authors have worked in this direction by proposing attack characterization models.

- In [14], authors exploit UML to describe eleven attacks observed in four layers. This approach lacks attack modeling on the application layer, no externally initiated attacks, nor clearly specified routing protocol.
- In [16], authors state the concept of a multi-layer attack and present two attacks in the form of sequence diagram.
- In [17], strategies to launch smart jamming attacks are presented. The authors design an UML sequence diagram to represent the jamming attack initiated from the inside only.

Table I summarizes a comparison of anti-malware proposals with advantages and disadvantages.

3. Problem Statement

In this section, the problem is defined and the consideration of new aspects is justified.

WSN are subject to various attacks that negatively affect decision-making. For example, they provide false information about events, they participate in disabling sensors, and they impersonate activities. We believe that such effects are the consequences of a concise exploitation by attackers of features related to sensor and WSN. So the wise step should be to study these features. The aforementioned works that have looked at these features is not very specific in terms of WSN layer coverage, attack typology and protocol typology. There is a need to study a larger number of attacks on all layers of the WSN while considering different aspects such as the position of the attacker, the protocols, the types of attacks, etc.

Existing solutions are partial due to the fact that authors provide knowledge only on smaller number of attacks and also because their focus neglect layers and aspects from which attacks succeed. The following explains why in this work, such aspects should be considered.

- Number of attacks: For sake of completeness, it is important to study more attacks than what have been studied in the aforementioned works. New attacks emerge relying on the earlier ones whereas other attacks are come with completely new strategies.

TABLE I. Comparison of proposals for attack countermeasures in WSN.

Proposal categories	Advantages	Disadvantages
1. Cryptography-based mechanisms [18], [19]	They are used to guarantee Confidentiality, Integrity, Availability (CIA) based on cryptography	Because of the high energy consumption of the sensors, these methods only apply to one or two attacks.
2. Intrusion detection approaches [20]–[22]	This second line of defense makes it possible to prevent malicious infiltration and uncover behavioral anomalies that are usually the sign of compromising nodes.	Authors require enough knowledge about attacks to design detection rules. Knowledge should be updated with the evolving behavior and discovery of attacks.
3. Solutions oriented to the characterization of attacks [14], [16], [17]	This solution provides an understanding of attack strategies to enable the design of more secure countermeasures.	No characterization of the attack on the application layer, no characterization of the attack on the collection of information, falsification of the nodes, no characterization based on routing protocol such as AODV.

- Components exploited: Depending on the attacks, one should study which components (physical, logical or human) are involved in the process represented in the UML sequence diagram. It is helpful to understand stepwise process and elements in which to infer protection. The previous studies suffer from this situation.

- Type of attacks: There are active and passive attacks. The first type is the only one considered in existing works. However, there are sniffing attacks and other attacks which run on the target to passively collect information. Generally, passive attacks are exploited by active attacks. It is so forth, important to also study passive attacks.

- Protocol used: The investigation of protocols is useful because different attacks spoof identity related messages exchanged to succeed. MAC and TCP are studied in the previous works. But we note that routing protocols such as Ad hoc On Demand Distance Vector (AODV) are not considered in these studies.

- Position of the attacker: The requests that are emitted by the attacker are different depending on whether they come from inside or from outside. For instance, the attack can be initiated by a node and an attack can be initiated wirelessly by a human. Only attacks launched from inside are stated by authors. Strategies related to attacks from outside are also of importance.

- Layer concerned: Features from WSN layers are exploited by attackers to succeed their activities. An attack can be represented as the composition of features from different layers. However, authors have only looked at some layers. Considering the emerging attacks, the unconsidered layers are exploited. More, some attacks are cross-layer meaning that they exploit all the layers to deploy their actions.

4. Characterization of attacks

A dataset of sixteen attacks is our focus. The selection is based on research in literature. We have identified the most popular attacks found in related researches. Moreover, the selection has been made because and the ones on which

WSNs connects small devices, sensors, and a base station. Sensors wirelessly communicate with protocols described in IEEE 802.11 [23]. According to [24], there are several types of WSN such as underwater, underground, multimedia, mobile and terrestrial. This paper deals with the latter.

In this section, the sequence of activities for sixteen attacks are designed based on UML. The main WSN attacks are grouped by layers. The objective in the layer is provided, then the attacks falling in this layer are formalized as previously stated. The sequence diagram is adapted by specifying layers and components involved during subsequences of the whole process. In each timely step, the logical and physical components are emphasized.

A. Physical layer

This layer provides a transmission channel for binary flows using frequency selection, carrier frequency generation, signal deviation, modulation and data ciphering [25]. The following attacks target this layer: Eavesdropping, node tampering and basic jamming.

1) Eavesdropping

Eavesdropping is a fundamental precondition for many other attacks. It is an attack in which the attacker intercepts radio signals without destroying their integrity [25]. This attack affects mainly confidentiality. This attack is shown in Figure 3.

The following explains each step of this attack.

1) Monitors transmission

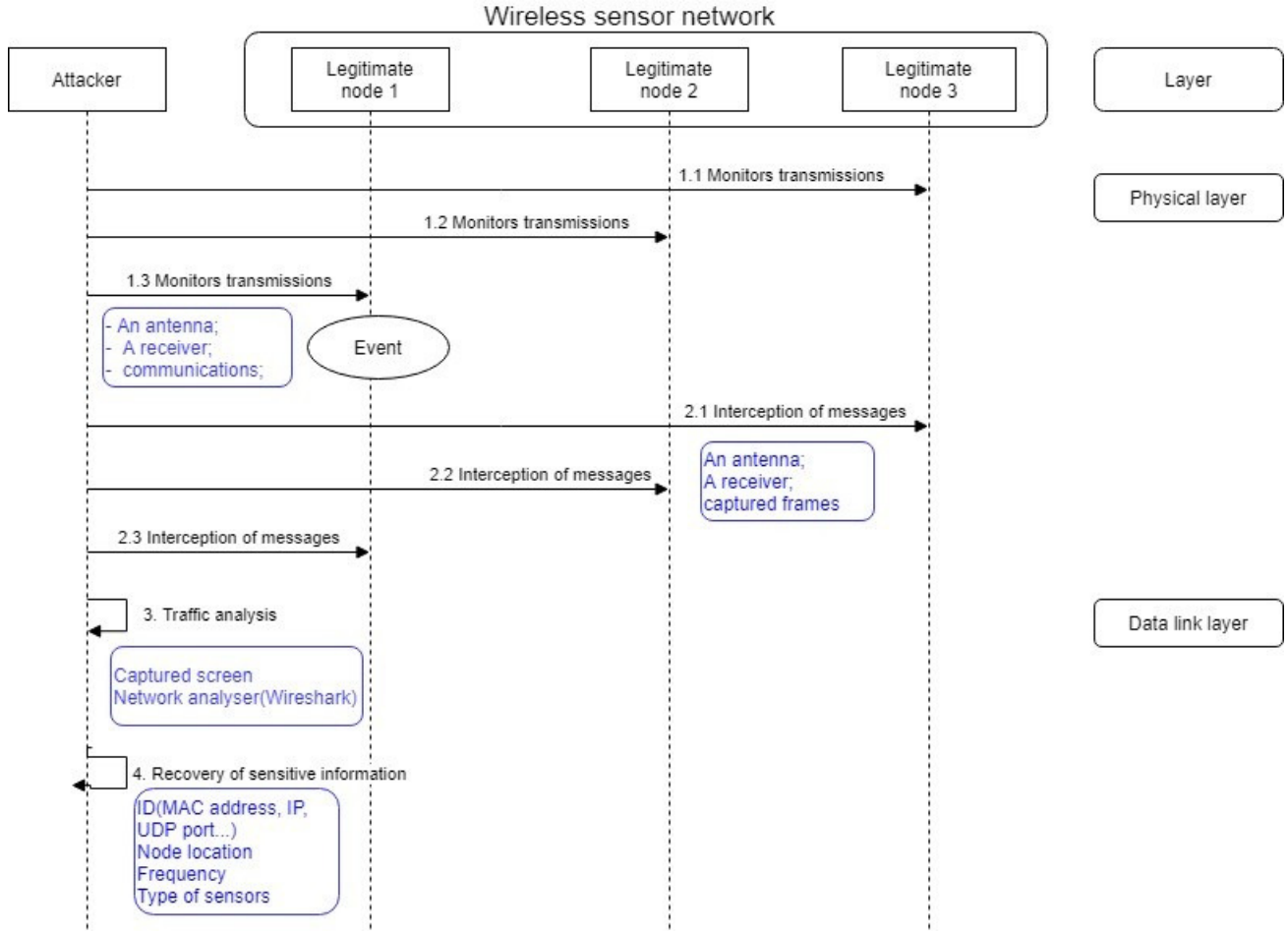

Figure 3. UML sequence diagram for attack gathering information.

- The attacker stands outside to monitor communications (electromagnetic waves) within the network. An antenna and a receiver are exploited.

2) Interception of messages
   - The triggering of an event in the network generates traffic.
   - The attacker captures all packets exchanged between nodes and saves the captured frames to a file (e.g. .pcap).
3) Traffic analysis
   - The attacker performs an analysis of the captured frame file using a network analyzer (Wireshark).
4) Recovery of sensitive information
   - Result of packet interception and traffic analysis (ID, Frequency, Type of sensors, Topology, Node location).

2) Node tampering attack

A WSN is generally deployed in hostile environmental conditions and far away. In this context, the distributed and unmonitored nature of the deployment of WSN makes them vulnerable to physical attack. The attacker physically destroys the node, impedes related circuits, educes cryptographic resources and alters the sensors's codes [26]. This attack is designed in Figure 4.

The following explains each step of this attack.

1) Collect_informations ()
   - Function that allows collecting information on the target (Frequency, key, ID...) by means of the attack information gathering (eavesdropping and traffic analysis).
2) Initiation physical capture
   - The attacker performs an extraction of the network node by moving through the sensor deployment environment.
3) Extraction of sensitive information
   - Using a computer, the attacker accesses the information contained in the given memory of the node.
   - The attacker tries to retrieve sensitive infor-

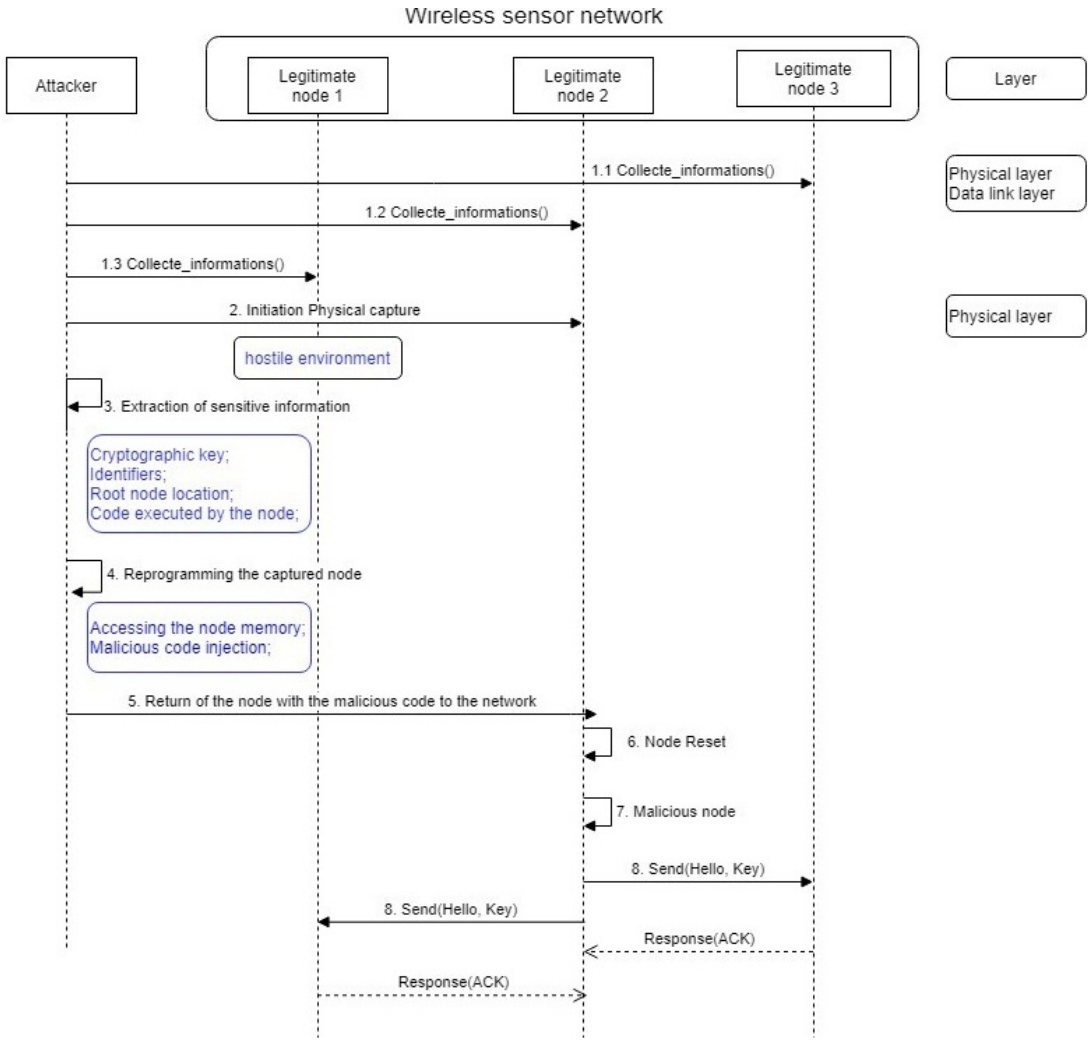

Figure 4. UML sequence diagram for node tampering attack.

mation (ID, Cryptographic key, MAC address, code executed by the Node).

4) Reprogramming the capture node
   - The attacker also accesses the node's flash memory;
   - Node source code modification or malicious code injection.
5) Return of the node with the malicious code to the network
6) Node reset
   - The attacker executes the remote node reset function (Bootloader) to be able to execute the previously injected malicious code.
7) Malicous node
   - Node diverted from its main objective by an attacker
8) Send(Hello, Key)
   - Use the Neighbour Discovery protocol by sending a Hello Include cryptographic keys message to all these neighbours.
   - Updating the malicious node's neighbour table
9) Response(ACK)
   - Response of nodes within range of the malicious node;
   - Updating the neighborhood table of the legitimate node within reach of the malicious node.

3) Basic Jamming attack

Basic jamming attack makes use of electromagnetic energy to intermeddle or disrupt exchanges between legitimate nodes [25]. Here, the attacker sends radio signals to interrupt data communication [27]. Authors in [27] have divided jamming attacks into four taxonomies: constant, deceptive, random and reactive jamming. This attack is designed in Figure 5.

The following explains each step of this attack.

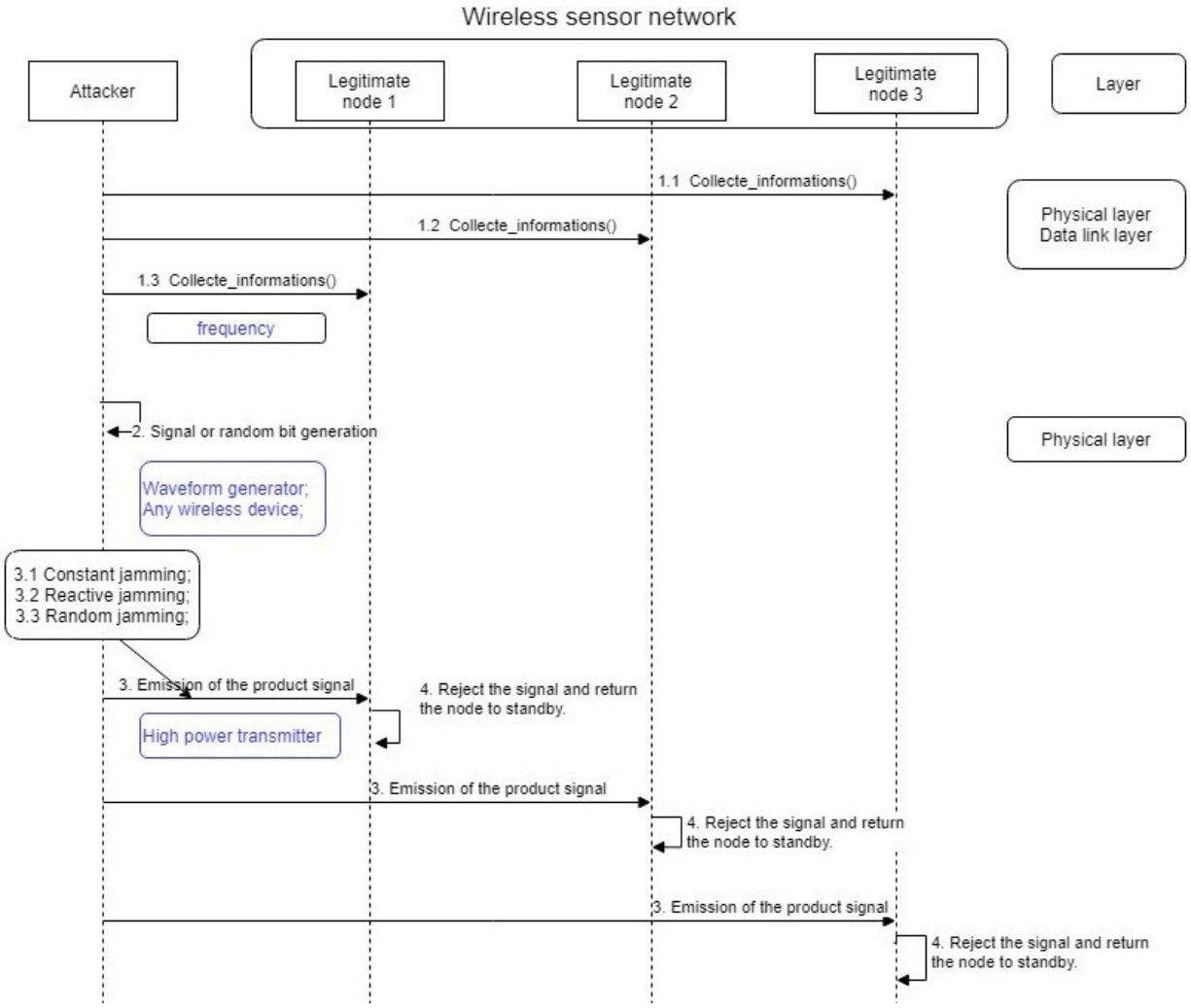

Figure 5. UML sequence diagram for basic jamming attack

1) Collect_informations()
   - Function that allows collecting information on the target (Frequency, Key, ID) by means of the attack information gathering (eavesdropping and traffic analysis).
2) Signal or random bit generation
   - Using a generator or wireless device, the attacker produces a signal or noise able to disrupt the network once injected with a slightly higher power.
3) Emission of the product signal
   - Based on the recovered frequency, the attacker transmits the resulting signal/noise using a high-power transmitter, an antenna and one of the jamming strategies.
4) Reject of the signal and return the node to standby
   - After receiving the signal/noise from at least one of the nodes, the node will simply reject the signal and go into standby mode. It will wake up from time to time to check if the signal is still present in the network.
   - If the attacker maintains good timing he will be able to block the communication channels of the target nodes and thus create a DoS.

B. Data link layer

In this layer, a reliable channel for communication is provided to neighboring nodes. It also allows, most often, to detect and possibly correct certain errors occurring in the physical layer (when the electromagnetic signal degrades) [23], [25]. Attacks on the WSN's data link layer generally consist of four types of attacks: collision, intelligent jamming, traffic analysis and sleep privation.

1) Collision

Any attempt of simultaneous transmission between two nodes on the same frequency is called collision. The direct consequence is packet corruption and disruptions in the operation of the network. In a collision attack, an attacking node does not rely on the access control protocol and collides with with noisy packet [25]. This attack is designed in Figure 6.

The following explains each step of this attack.

1) Compromise_Node()
   - A function that allows a node to be compromised and taken over by means of either a tampering attack on the node or a code injection attack.
2) Malicious node
   - Node diverted from its main objective by an attacker
3) Collision initiation
4) Send(RTS)
   - Based on the MAC protocol, the source node sends a "Request to Send" (RTS) packet to check the availability of the channel of its neighboring node and thus make the channel reservation.
5) Send (CTS)
   - After receiving the RTS, the target node sends back a "Clear to Send" (CTS) packet just after a short inter-frame space (SIFS) to acknowledge readiness for request reception.
   - The RTS and CTS packets specify how long the channel is occupied so that neighboring nodes can update their Network Allocation Vector (NAV).
6) Collision
   - The malicious node ignores the busy signal of the destination node and sends a corrupted packet creating a collision and consequently the rejection/deformation of the data sent by the source node.

2) Intelligent jamming attack

This attack transfers data packets under known protocol rules, interfering with communications and consuming node power. Only deceptive jammers, random jammers and reactive jammers can launch the smart jamming attack when they transmit regular packets to sensor nodes [13]. This attack is formulated in Figure 7.

The following explains each step of this attack.

1) Collecte_informations ()
   - Function that allows collecting information on the target (Frequency, key, ID...) by means of the attack information gathering (eavesdropping and traffic analysis).
2) Signal or random bit generation
   - Using a shape generator or wireless device, the attacker produces a signal or noise able to disrupt the network once injected into it with a slightly higher power.
3) Send(RTS)
   - Based on the MAC protocol, the source node sends a RTS packet to check the availability of the channel of its neighboring node and thus make the channel reservation.
4) Emission of the product signal
   - Based on the recovered frequency, the attacker transmits the resulting signal/noise using a high-power transmitter, an antenna and one of the jamming strategies.
5) Send(CTS) of the victim node
6) Collision
   - The purpose of this broadcast is to create a collision on the victim's CTS frame based on SIFS.

3) Traffic analysis

Encrypted messages do not prevent any furtive analysis of communication patterns of the WSN. Sensor communication activities can, by their nature, reveal enough information to the adversary and facilitate damage in the sensor network [17]. An example of this attack is shown in Figure 3.

4) Sleep privation attack

Here, the attacker simulates benign interaction with the victim to maintain the victim's node spoiling energy. For instance, RTS and CTS messages are periodically sent to the victim maintaining it to accept requests. As a result, the victim's lifespan is considerably reduced. This attack is difficult to detect because it is only achieved via seemingly innocent interactions [5]. This attack is designed in Figure 8.

The following explains each step of this attack.

1) Compromise_Node()
   - A function that allows a node to be compromised and taken over by means of either a tampering attack on the node or a code injection attack.
2) Malicious node
   - Node diverted from its main objective by an attacker
3) Initiation privation
   - This operation initializes the privation.
4) Send(RTS)
   - Malicious node forges RTS packets
   - Continuous sending of RTS packets to neighbouring nodes
5) Send (CTS)

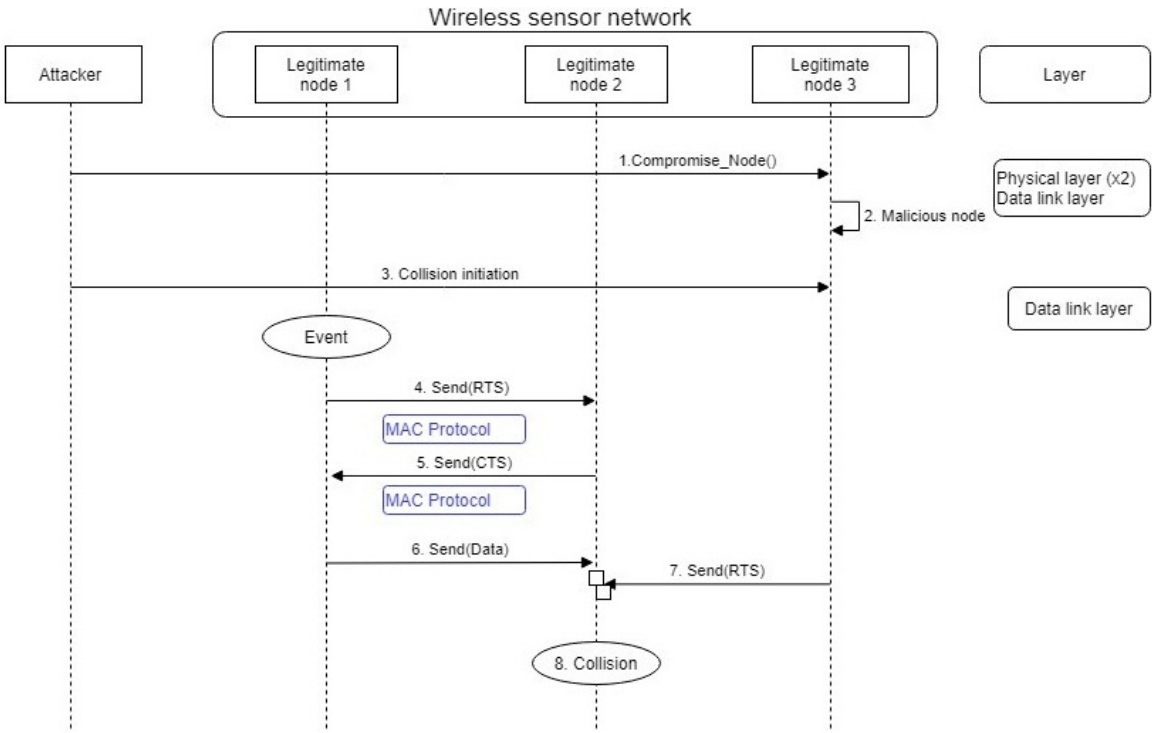

Figure 6. UML sequence diagram for basic jamming attack

- Neighbouring nodes will respond to fake RTS packets by sending CTS and thus remain in listening mode waiting for the data coming from the malicious node

6) Send(Data)
   - The malicious node will not deliver the data to the neighbouring node.

C. Network layer

The network layer is used to manage routing processes of data between nodes. The established route must optimize the energy consumed by the sensors and the latency times for data transport. This layer also defines the addressing process [28]. Attacks on the WSN's Network layer generally consist of six types of attacks: hello flooding, sinkhole, blackhole, greyhole, wormhole and Sybil.

1) Hello Flooding

The management of most communication protocols requires periodic exchange of hello packets. Indeed, acknowledgement of such packets confirms that the sender lives within its radio coverage area. An attacker therefore uses a high-powered transmitter to mislead many nodes to trust it [29]. In so doing, benign nodes will send their data to the fake node. This attack is designed in Figure 9.

The following explains each step of this attack.

1) Collecte_informations ()
   - Function which allows collecting information on the target (Frequency, Key, ID...) by means of the attack information gathering (eavesdropping and traffic analysis).
2) Forging packets (Hello)
   - The Hello packet includes: ID, Node location
3) Send (Hello)
   - Using a high-powered transmitter, the attacker sends Hello packets to the nodes
   - Based on the neighbour discovery mechanism, the works present in the attacker's transmission zone will be able to receive these packets
4) Updating of the neighbourhood table
   - The nodes will update their neighborhood table with a fake neighbour list

2) Sinkhole attack

A sinkhole attack aims at gathering information by transferring all traffic to a fake node that the attacker has put in an area. These attacks make a compromised node attractive to bounding nodes to the routing algorithm [30]. There are two variants of this attack: black hole and grey hole. An example of this attack is shown in Figure 10.

The following explains each step of this attack.

1) Compromise_Node()
   - A function that allows a node to be compromised and taken over by means of either a tampering attack on the node or a code injection attack.

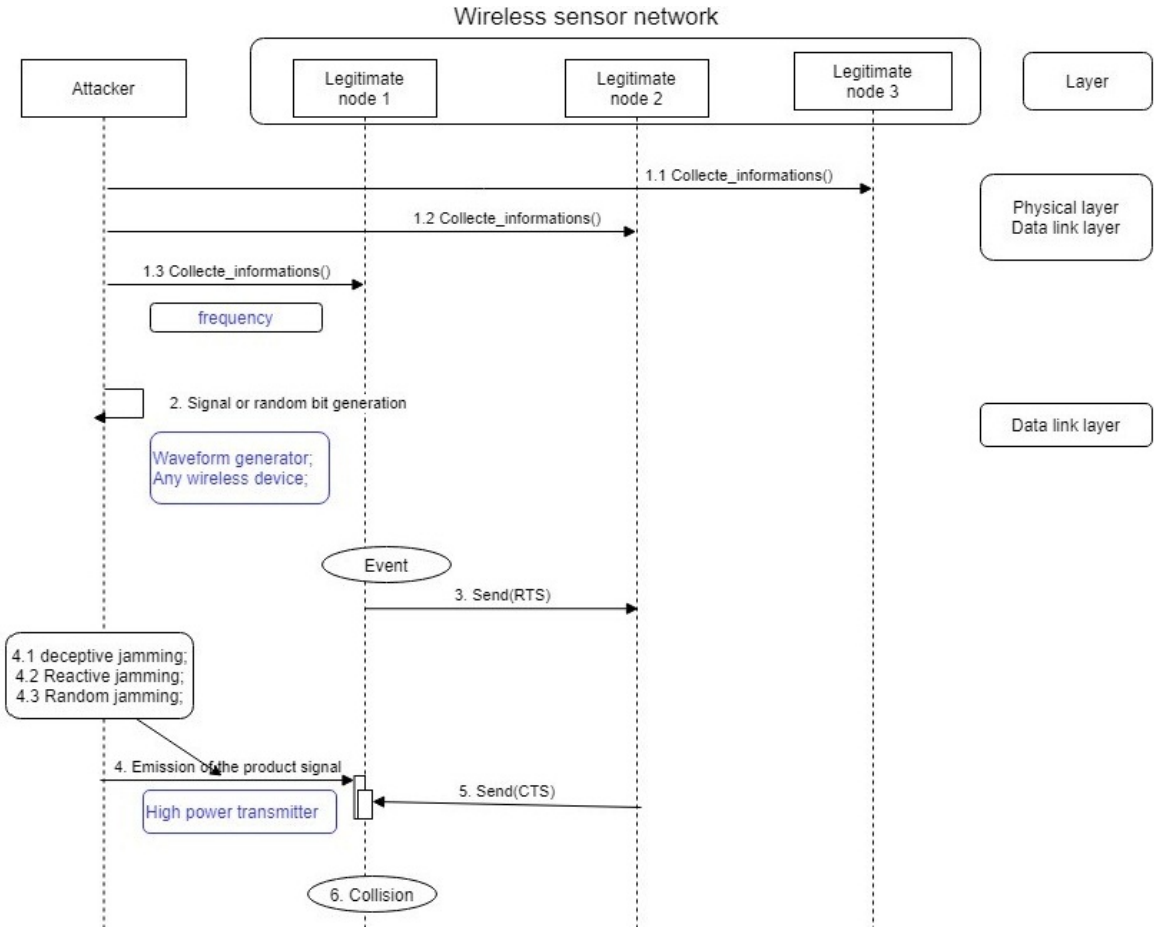

Figure 7. UML sequence diagram for intelligent jamming attack

2) Malicious node
   - Node diverted from its main objective by an attacker
3) Initiation sinkhole
4) Send(RREQ)
   - The source node broadcasts the RREQ packet to all nodes in the network.
   - The RREQ contains: The source and destination address, A sequence number
5) Send (RREP)
   - Upon receipt of RREQ by the destination node, the latter sends back a Route Reply (RREP) packet.
   - The message RREP contains: the address of all the Nodes making up the route, a sequence number, and the total number of jumps once the packet reaches the source node.
6) Modifcation of the RREP
   - Once the malicious node is reached, the RREP packet will be modified (number of high sequences, low number of jumps)
7) Send(RREP) modified
   - The malicious node sends the modified RREP to the source node.
8) Send(RREP)
   - The legitimate node sends the RREP with the real values contained in it.
9) Selecting of the RREP by comparing according to the sequence number
   - The source node receives the RREPs (modified and unmodified) and makes a comparison by referring to the number sequence and the number of jumps contained in these packets.
   - Choose the recent RREP, i.e. the one with the highest sequence number and the lowest number of jumps.
   - Rejection of all other RREPs
10) Send(Data)
   - Sending data that will all be directed to the malicious node
11) Malicious node attracts trafic

3) Blackhole attack

In the black hole attack, the malicious node absorbs all packets without transferring them to their destination [25].

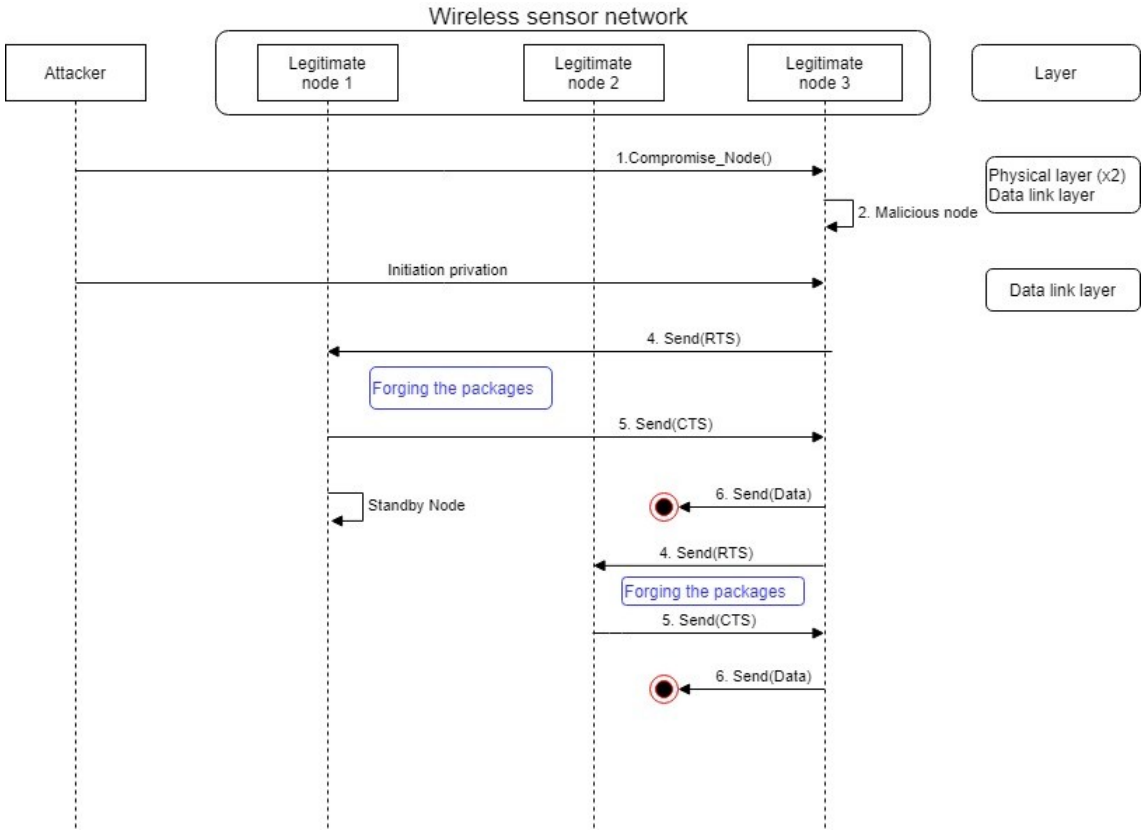

Figure 8. UML sequence diagram for sleep privation attack

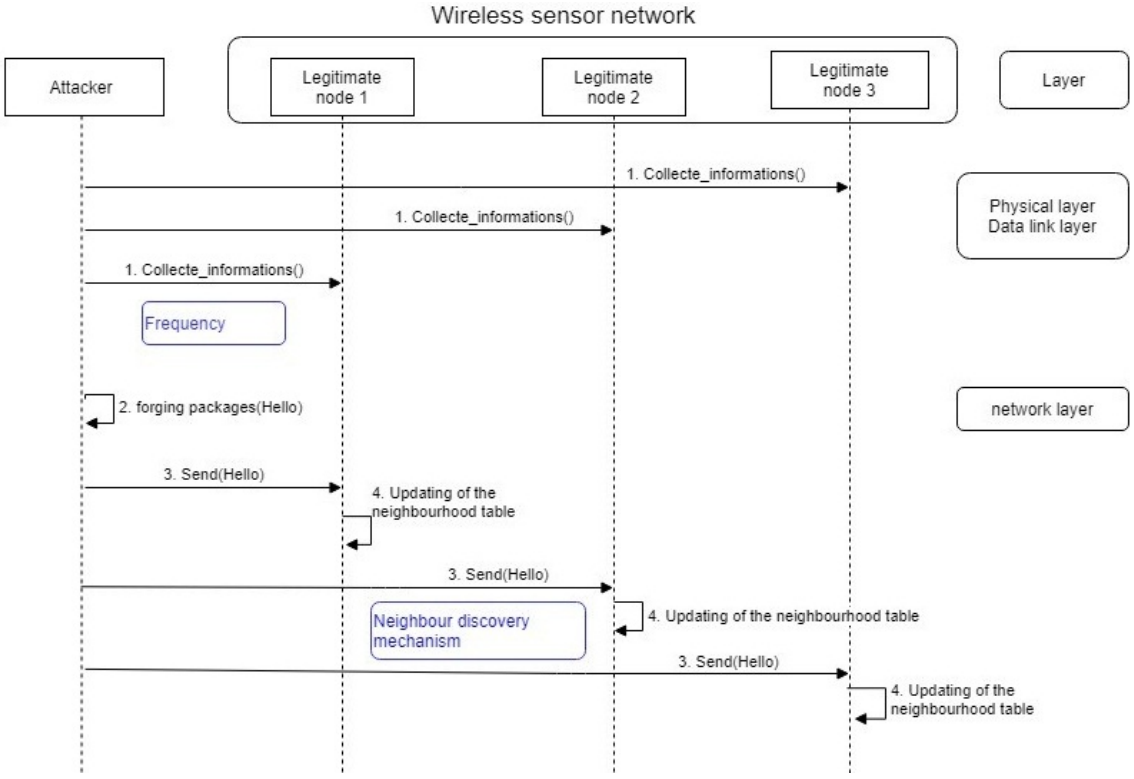

Figure 9. UML sequence diagram for hello flood attack

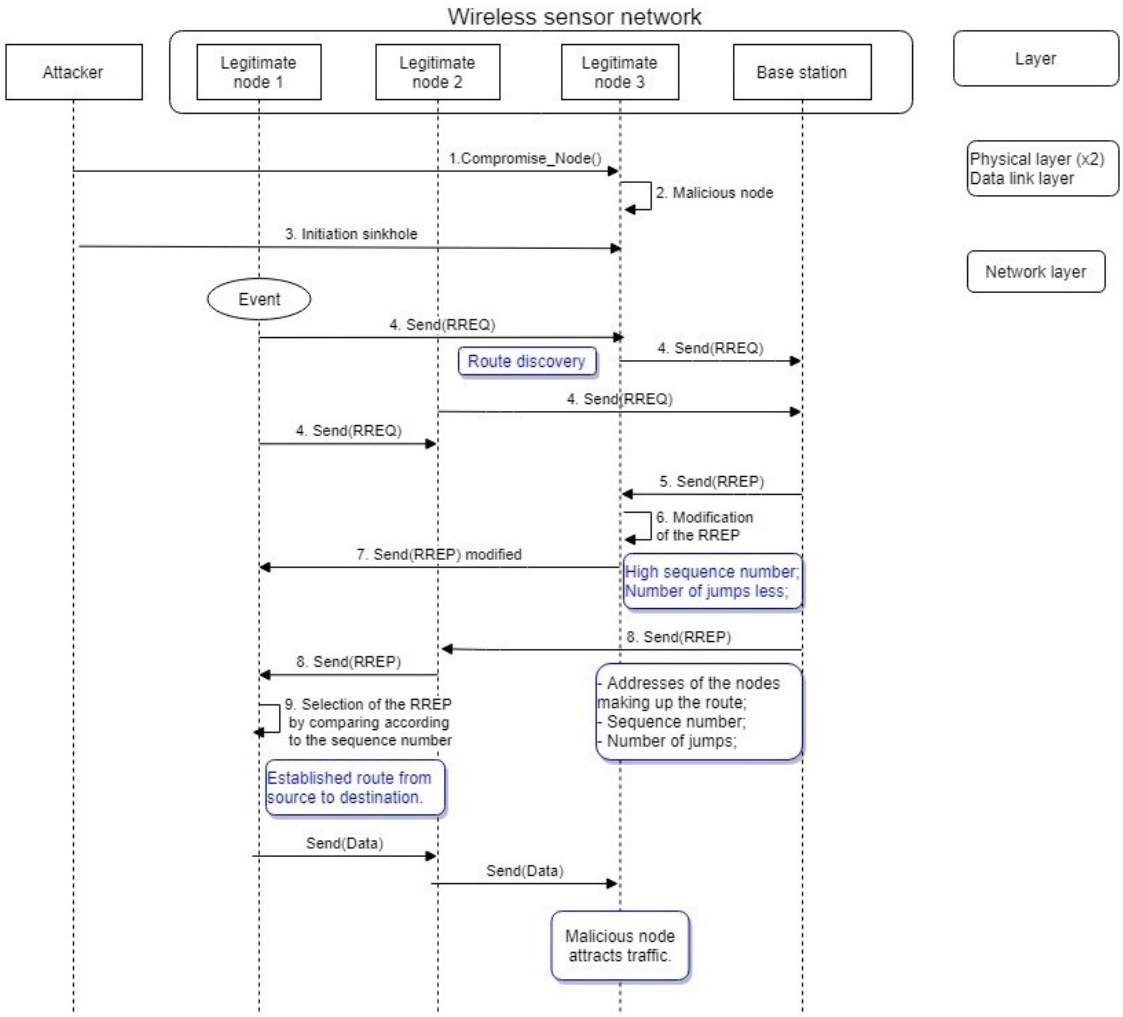

Figure 10. UML Sequence Diagram for Sinkhole attack.

This attack is designed as in Figure 11.

The following explains each step of this attack.

1) Compromise_Node()
   - A function that allows a node to be compromised and taken over by means of either a tampering attack on the node or a code injection attack.
2) Malicious node
   - Node diverted from its main objective by an attacker
3) Sinkhole()
   - This is an operation to attract traffic by sending false routing information to neighboring nodes by means of a sinkhole attack.
4) Initiation blackhole
5) Send(Data)
   - Sending data that will all be directed to the malicious node
6) Malicious node drop all traffic
   - All the data that will pass through the malicious node will be completely deleted.

4) Greyhole attack

Unlike blackhole attack, the grey hole shares certain information. For instance, it relays all routing information and blocks critical information. This type of attack is therefore hard to identify, because the evil node does not remove all received messages [19]. This attack is designed as in Figure 12.

The following explains each step of this attack.

1) Compromise_Node()
   - A function that allows a node to be compromised and taken over by means of either

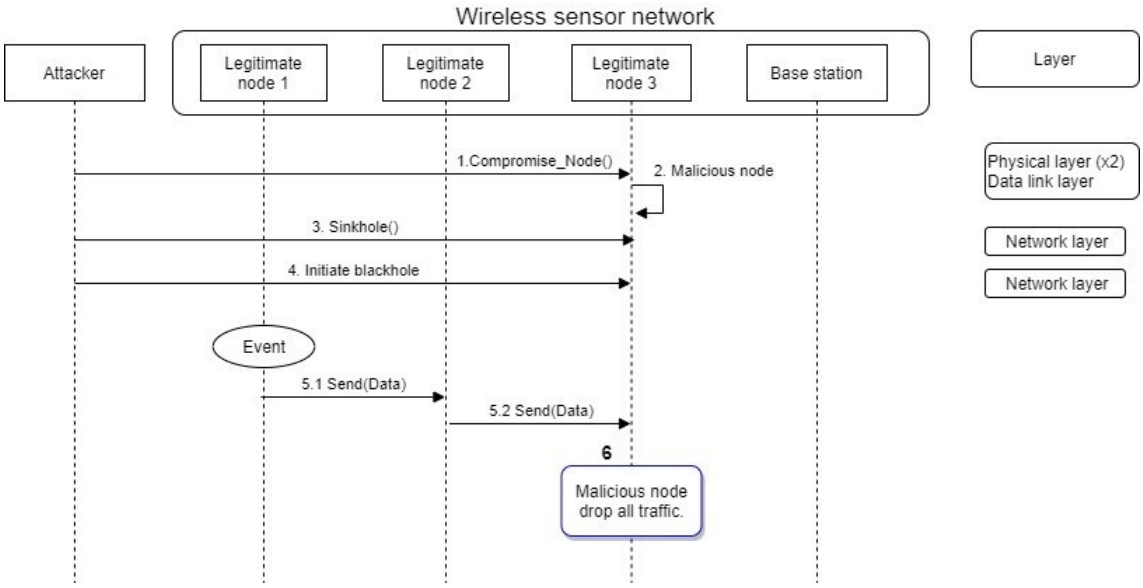

Figure 11. UML sequence diagram for blackhole attack

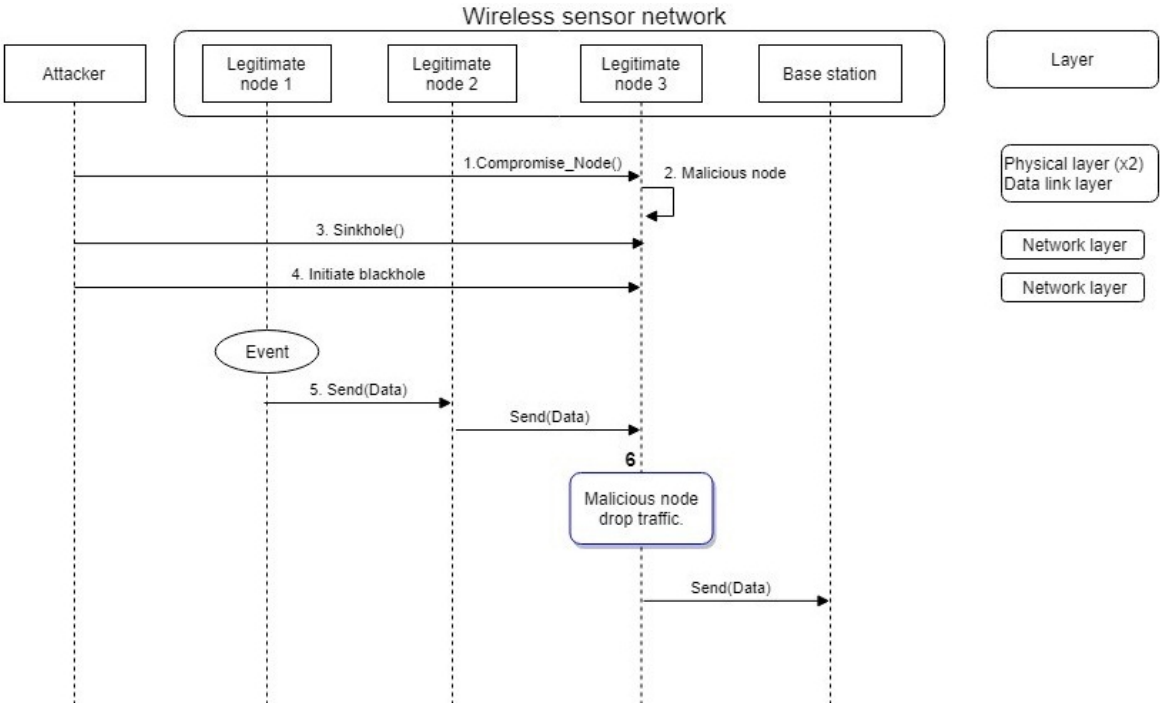

Figure 12. UML sequence diagram for greyhole attack.

a tampering attack on the node or a code injection attack.

2) Malicious node
   - Node diverted from its main objective by an attacker
3) Sinkhole()
   - This is an operation to attract traffic by sending false routing information to neighboring nodes by means of a sinkhole attack.
4) Initiation greyhole
5) Send(Data)
   - Sending data that will all be directed to the malicious node
6) Malicious node drop all traffic
   - All data that will pass through the malicious node will be selectively transferred to the next node or base station.

5) Wormhole attack

In this attack [29], a malicious node captures packets from its juxtaposed nodes and forwards them to another malicious node, which is a node to relay these packets. This fact can deform the distances between nodes and mislead discovering of neighbours. As a result, a sensor node mistakenly selects a remote node as its closest neighbour for next transmission. This results in a rapid depletion of resources and a reduction of life of the network. This attack is presented as in Figure 13.

The following explains each step of this attack.

1) Compromise_Node()
   - A function that allows a node to be compromised and taken over by means of either a tampering attack on the node or a code injection attack.
2) Sinkhole()
   - This is an operation to attract traffic by sending false routing information to neighboring nodes by means of a sinkhole attack.
3) Initiation wormhole
4) Wormhole link
   - Auxiliary channel for long-distance, low-latency (wireless) transmission of different network communication channels
5) Send (Data) - Packet capture - Sending the captured packets through the wormhole link
   - The malicious node_1 captures the data from one area and injects it into a remote area via the Wormhole Link, making the nodes believe located at the ends (next to the malicious node_2) that they are neighbours. This attack is exploited to support other attacks including selective transmission.

6) Sybil attack

The sybil attack consists of a corrupt agent spoofing identities of several nodes in the network [23]. These identities concern:

- Invented, of non-existent nodes;
- Existing nodes, but distant from the corrupted node;
- Nodes that are destroyed and virtually replaced by the corrupted node.

Thus, the attacking node relies on these usurped identities to be elected as cluster head and to create fake routing paths [5]. This attack is formulated in Figure 14.

The following explains each step of this attack.

1) Compromise_Node()
   - A function that allows a node to be compromised and taken over by means of either a tampering attack on the node or a code injection attack.
2) Malicious node
   - Node diverted from its main objective by an attacker
3) Initiation Sybil
   - The malicious node randomly generates IDs based on the size (number of bits) of the IDs of the legitimate nodes.
   - The malicious node steals the IDs of the legitimate nodes already in the network by performing ID replication.
4) Sybil node
   - It is a malicious node that has several IDs
5) Send (Hello, ID_1)
   - At each Send (Hello), the malicious node will use an di[U+FB00]érent ID to update the neighborhood table of neighboring nodes with fake IDs..

D. Transport layer

The transport layer manages the sequencing, the reorganization of the packets and flow control by specifying a reliable transport of the packets. Two examples of possible mechanisms in this layer are Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). [31]. The attacks in this layer consist of two types of attacks: flooding and desynchronization.

1) Flooding

Deny of Service (DoS) is one of the flooding attacks. An attacker sends many unnecessary packets to a legitimate node to prevent it from communicating normally and degrading the life of the network. For example, in a TCP SYN flooding attack, the attacker floods request packets for connection establishment to the victim. After reception, the

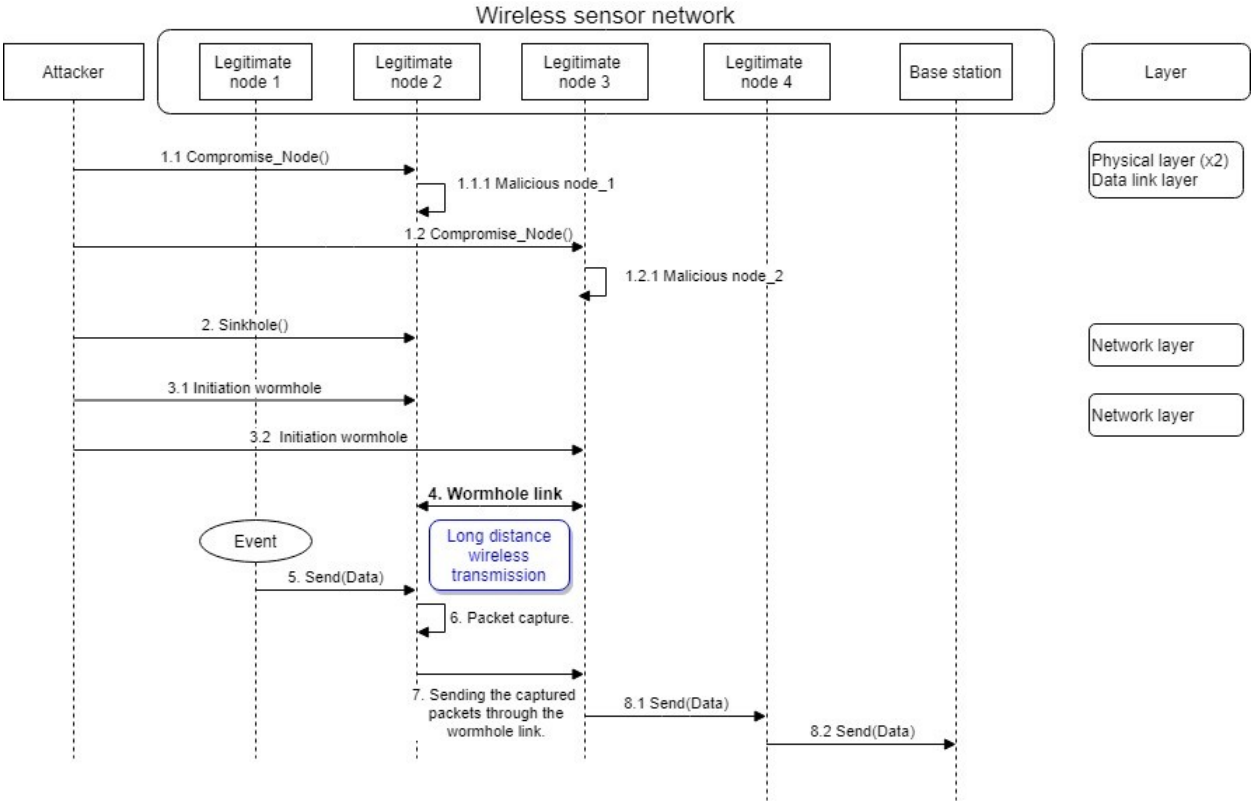

Figure 13. UML sequence diagram of wormhole attack.

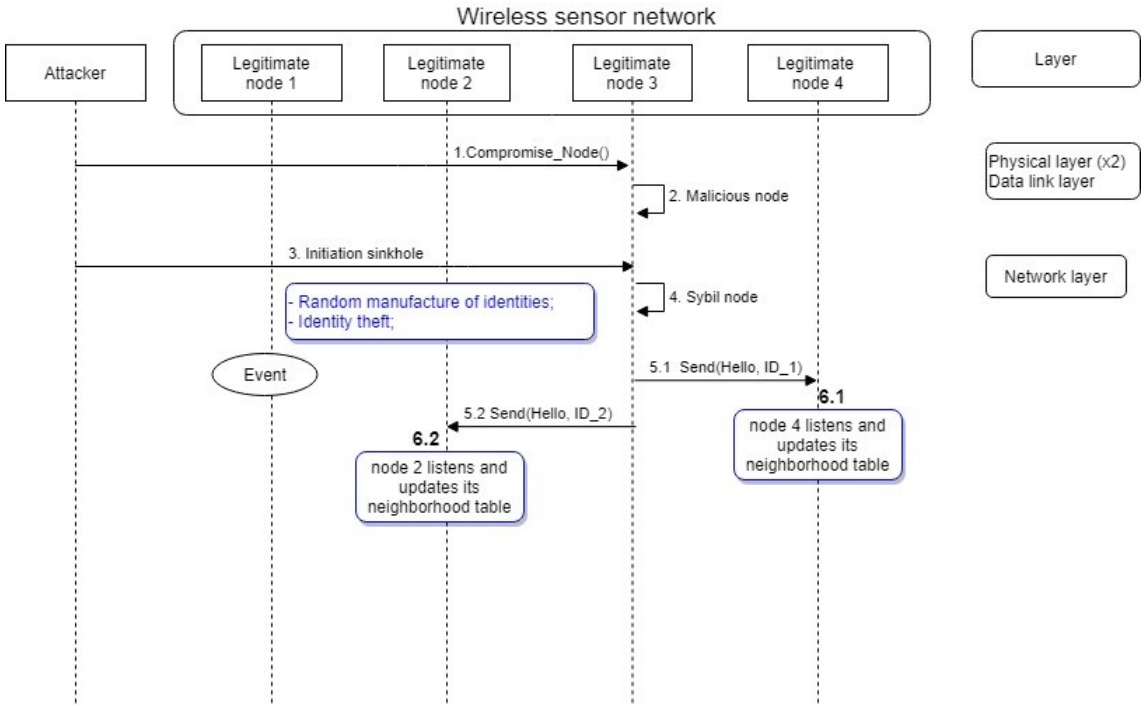

Figure 14. UML sequence diagram for Sybil attack.

victim replies to the requester's acknowledgement packets and waits for the connection. Additionally, the victim reserves storage space for transmission control. In this attack, normal operations are evacuated and resources are spoiled [25]. This attack is designed in Figure 15.

The following explains each step of this attack.

1) Compromise_Node()
   - A function that allows a node to be compromised and taken over by means of either a tampering attack on the node or a code injection attack.
2) Malicious node
   - Node diverted from its main objective by an attacker
3) Initiation flooding
4) Send(SYN)
   - The Malicious Node sends a SYN packet to the Victim Node to initiate the connection.
5) Send(SYN, ACK)
   - The victim node responds to the malicious node with a SYN + ACK packet.
   - The victim node reserves memory space for a future connection between itself and the malicious node.
6) Not Send(ACK)
   - The malicious node does not send an ACK packet to confirm the connection.
   - This operation will be repeated until the memory space is exhausted to prevent connections between legitimate nodes

2) Desynchronization

This type of attack disrupts the current connection. The attacker maliciously forces the final host to start retransmitting lost frames. This is done by repeatedly transmitting false messages comprising sequence numbers and control flags to the victims. If timing is correct, the attacker may hinder or degrade the ability of the final host(s) to effectively exchange or share data. This situation disrupts available connection [32]. This attack is designed in Figure 16.

The following explains each step of the attack.

1) Compromise_Node()
   - A function that allows a node to be compromised and taken over by means of either a tampering attack on the node or a code injection attack.
2) Malicious node
   - Node diverted from its main objective by an attacker
3) Initiation Desynchronisation
4) Sending fake packets
   - The attacker forges fake packets using: a sequence number, control flag.
   - The attacker repeatedly sends these fake packets to the victims.
5) Request for retransmission
   - After the reception of the fake packet by the Victim Nodes, a retransmission of the missing frames is requested by the victim node creating the exhaustion of energy.

E. Application layer

This layer aims at abstracting the main functions of the detection. This is done by leveraging software and hardware transparent to the final user. The application layer has multiple processes operating at the same time and manages user requests [33]. Attacks targeting this layer consist of code injection.

1) Code injection attack

The attacker embeds a worm into a node to disaggregate or take complete control of the node, which may reduce network capacity and perform its intended functions [25]. Authors in [34] have shown that it is possible to inject a worm which propagates via the wireless sensor network and possibly creates a sensor botnet. This attack is formulated in Figure 17

The following explains each step of this attack.

1) Collect_informations ()
   - This operation collects information on the target (Frequency, Key, ID...) by means of the attack information gathering (eavesdropping and traffic analysis).
2) Location of the node
   - Use location elements (Type of sensors, OS)
3) Vulnerability Scan
   - Based on the permanent addresses of the node, the attacker performs a scan to highlight software vulnerabilities (OS, more precisely the flood protocol)
4) Presence of vulnerabilities: returns yes or no.
5) Extraction of the OUI
   - The attacker relies on the MAC address of a node to find out about hardware vulnerabilities

From this characterization, another type of attack as presented by [16] emerges. It is an attack that uses the information of one layer to produce an attack on another layer, or to launch an attack on several layers in cooperation. This type of attack is due to the limitations of some countermeasures such as encryption [35]. This type of attack can be grouped into two categories: inter-layer attacks and attacks of connivance.

1) Attacks made of other layers

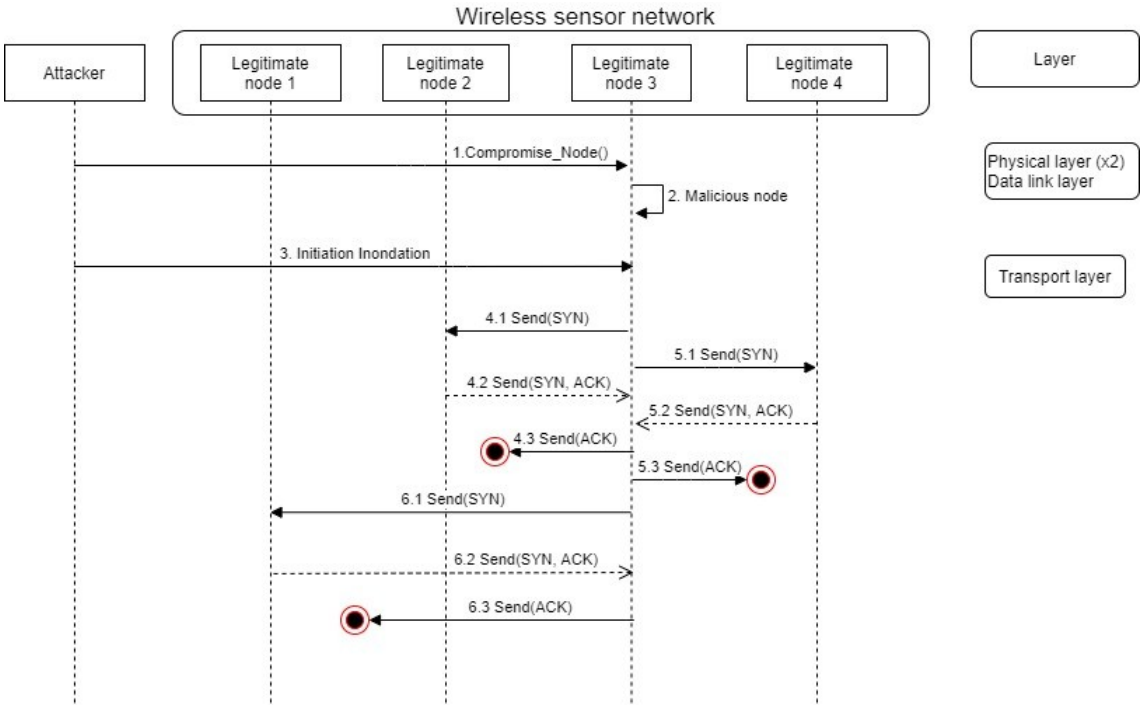

Figure 15. UML sequence diagram of flooding attack.

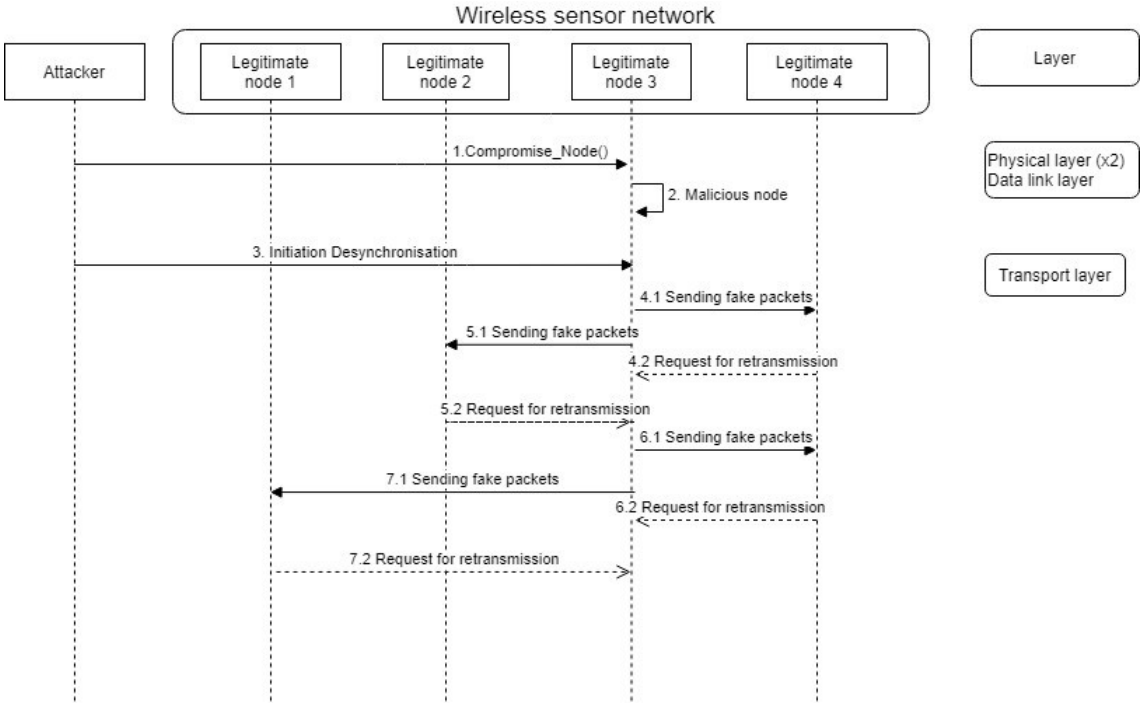

Figure 16. UML sequence diagram for the desynchronization attack.

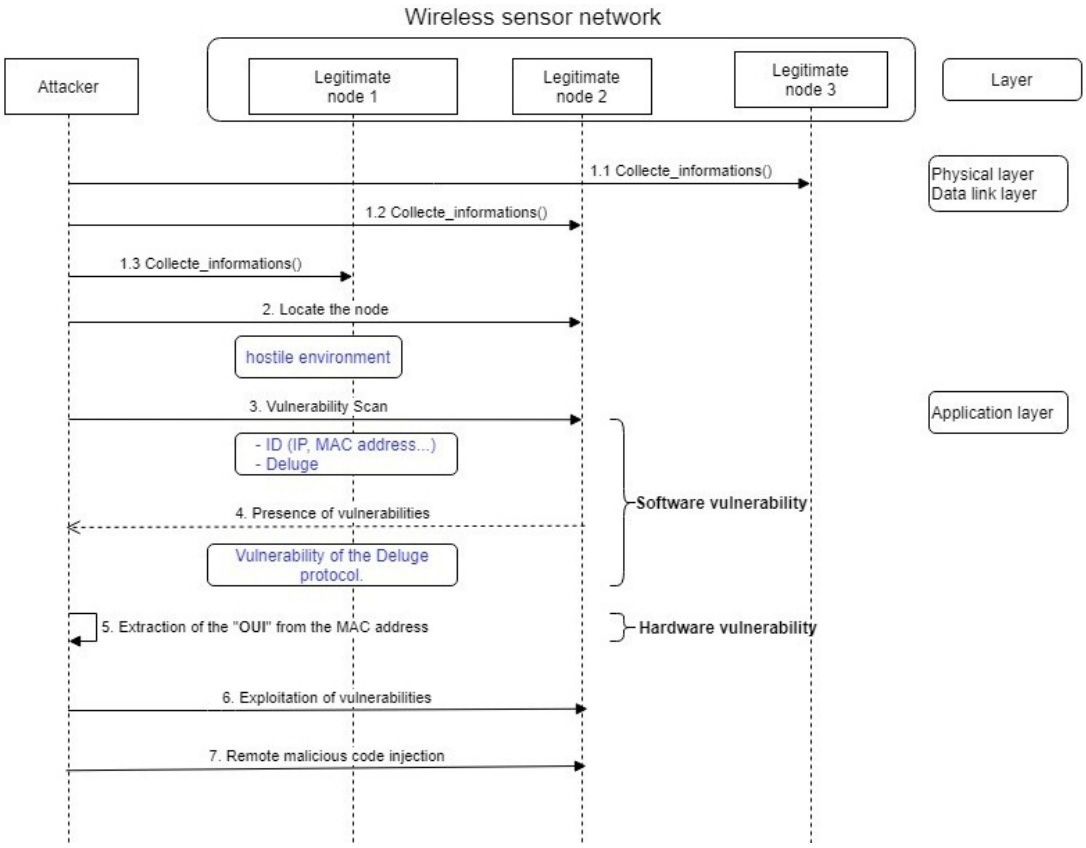

Figure 17. UML sequence diagram for the code injection attack.

Figure 18 describes the number of layers that an attack uses to achieve its goal. This study highlights the layer most commonly used in attacks within wireless sensor networks. For attacks initiated from inside, the compromise of the node can be physical (node tampering) or applicative (code injection).

- Attack related to gathering of information

They combine two types of attacks, each taking place on a specific layer: eavesdropping (physical layer) and traffic analysis (layer data link). All active attacks rely on data from the gathering information attacks (eavesdropping, traffic analysis): frequency, type of sensors, location, type of OS. These attacks certainly create the notion of inter-layer attacks.

- Hole attacks

These attacks have one thing in common which is the malicious node (physical or logical compromise) that attracts traffic by announcing false routing information. This false routing information is the product of a particular attack: the Sinkhole. This attack occurs on the network layer and exploits information from the "Information Gathering attack" and a node compromised to be initiated from inside. Thus, the sinkhole attack uses the combination of physical layer, link layer, physical or application layer and network layer.

2) Attacks of connivance

Another form of inter-layer attack is the connivance attack, which allows two or more attacks to be related. The two objectives are as follows:

- To escape from certain countermeasures. For example, a Sybil attack is in connivance with the flooding attack. This relation allows escaping ID authentication of node.
- To be more productive. Several attacks can be launched in connivance to be more productive. For instance: sinkhole and blackhole attacks; sinkhole and greyhole attack; sinkhole and wormhole attack; wormhole and blackhole attack; wormhole and greyhole attack.

F. Categorization of attacks

From the study based on the characterization of attacks, attacks can be grouped based on similarities and discrepancies.

1) Similarities

Tables II and III presents a categorization of attacks based on some similarity criteria [36]. They include fifteen

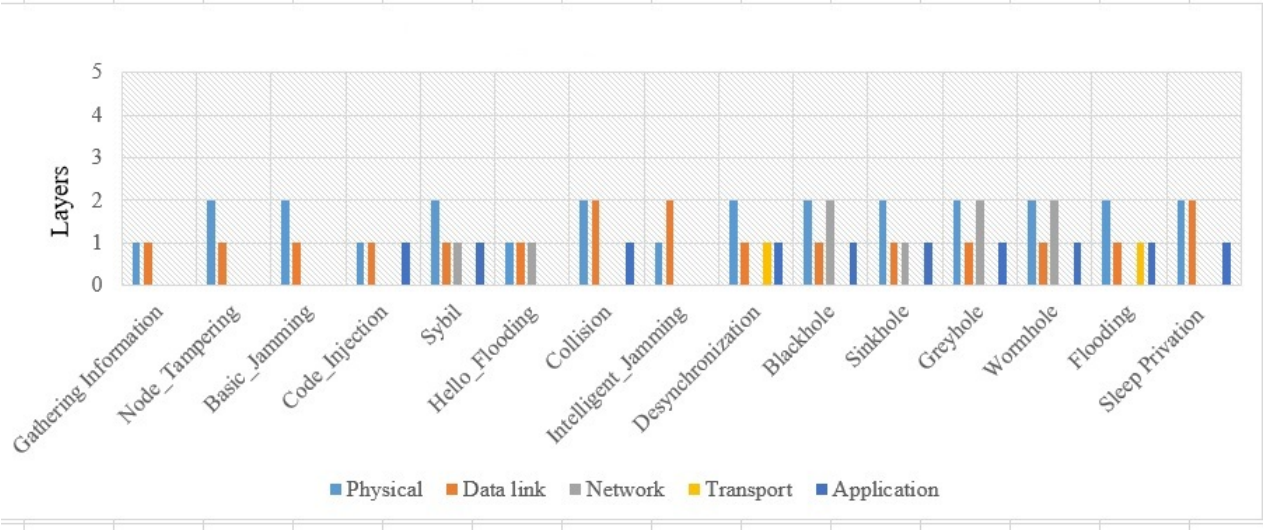

Figure 18. The different layers used in each attack.

lines and fifteen columns each representing an attack. The intersection of a row and column represents the similarity between these two attacks. We consider the row that represents "the information gathering" attack and the column that represents the "falsification of the node" attack. Their intersection is "theft of information" which represents their point of similarity.

2) Divergences

Concerning divergences, attacks can be summarized in several aspects.

- **Type of injected packets**

Depending on the nature of the attack, the malicious entity to be studied by considering the creation of scenarios based on collaboration diagrams to better represent the (node) injects a specific type of packet into the network. Depending on these attacks, there are the following types of packets (i) Basic interference: noise (ii) Intelligent Jamming: noise (iii) Collision: Data (iv) Hello Flood: Hello (v) Flood: SYN (vi) Desynchronization: Packet (vii) Sleep deprivation: RTS.

- **Packet removal mode**

This divergence is specific to the hole attacks (black hole and gray hole). The purpose of black hole is to completely remove packets whereas gray hole performs a selective packet removal.

- **Layers**

The protocol stack of WSN includes five layers. Each layer is subject to a particular attack to be initiated.

TABLE IV. Comparison with related studies

Subjects covered	[14]	[17]	[16]	This work
Number of attacks	11	Jamming (04 strategies)	2	16
Multi-layer appearance between attacks	No	No	Yes	Yes
Types of attacks	Active	Active	Active	Passive and Active
Protocol used	MAC, TCP	MAC, TCP	MAC, TCP	MAC, TCP, AODV
Position of the attacker	Internal only	Internal only	Internal only	Internal and External
Layer concerned	physical, data link, network and transport	physical and data link	physical, data link, network and transport	All the five layers
Components involved	No	No	No	Yes

TABLE II. Attacks by similarities

Attacks	Information gathering	Node tampering	Basic jamming	Collision	Intelligent jamming	Hello flooding	Flooding
1.Information gathering	-	Stealing information	Using collected information	Using collected information	using collected information	Using collected information	Using collected information
2.Node tampering	layer	-	layer	Malicious node			Malicious node
3. Basic jamming	layer	layer					
4. Collision	layer		noise injection		layer, goal attack		
5. Intelligent jamming	layer		noise injection				
6. Hello flooding							fake packet injection
7. Flooding						fake packet injection	
8. Desynchronization						fake packet injection	fake packet injection
9. Sinkhole						layer	
10. Code injection		firmware modification					
11. Blackhole				packet loss	packet loss	layer	
12. Greyhole				packet loss	packet loss	layer	
13. Wormhole						layer	
14. Sybil						layer	
15. Sleep privation				layer			

- **Type of attacks**

There are two broad categories of attacks: passive attacks and active attacks. According to Mohammadi et al. [37], passive attacks concern privacy (listening to, collecting and pilfering information by capturing data exchanges or by observing packets exchanged within a WSN); active attacks operate actions such as inserting faulty data, pretending, altering resource and data flows, making holes in security protocols, destroying network nodes, sensors, performance degradation, disruption of functionality and network overload. Thus, active attacks include the following: basic jamming, node tampering, collision, smart jamming, sleep deprivation, sybil, black hole, gray hole, sinkhole attack, Hello flood, wormhole, flood, desynchronization, and code injection.

5. Comparison with similar works

Table IV presents some criteria for evaluating similar research against our approach. Seven criteria are used. The first concerns the number of attacks modelled. The second represents the number of layers used in a final attack. The third criterion represents the type of attack modelled. The fourth criterion indicates the protocol types of the data link, routing and transport layers used during the attacks. The fifth criterion indicates the position of the attacker (Malicious entity) when the attack is initiated. The sixth criteria represents the layers affected by the attacks. The seventh criteria specifies whether authors investigates physical and logical components involved during the attack process.

We note that research on attack characterization is based only on attacks initiated from the inside and on a few layers of the WSN. The authors propose a characterization of attacks at the level of WSN considering that the malicious entity is already present within the Network. This means that they do not consider fundamental attacks such as information gathering, node forgery or code injection. To make attacks more productive, authors in [16] present cross-

layer and conspiracy attacks. Unlike, our proposal provides formalizes sixteen attacks (two passive and fourteen active attacks) which use all the layers of the protocol stack (physical, data link, routing, transport and application) including the study of some protocols such as: MAC protocol, TCP and routing protocol AODV during attack scenarios. Cases of attacks initiated from inside and outside are modelled as well as attacks fundamental to other attacks (information gathering and node compromise: node tampering and code injection) to insert a malicious node into the WSN are presented. From the above, it is obvious that this research is com-plementary to the others and fundamental for the compre-hension and characterization of attacks in WSN. It is more fine-grained since it adds many other aspects such as the components being required, for putting in place detection mechanisms. However, for the sake of completeness, this dissection should be exhaustive since new attacks appear gradually. Additionally, this work does not elucidate the logic of processes behind attacks. It means that, as it is, one cannot directly exploit this result into a detection system.

TABLE III. Attacks by similarities (continued)

Attacks	Desynchroni-zation	Sinkhole	Code injection	Blackhole	Greyhole	Wormhole	Sybil	Sleep privation
1.Information gathering	Used the information collected	Using collected information	Using collected information	Using collected information	using collected information	Using collected information	Using collected information	Using collected information
2.Node tampering	Malicious node	Malicious node	firmware modification	Malicious node	Malicious node	Malicious node	Malicious node	Malicious node
3. Basic jamming								
4. Collision								layer, energy consumption
5. Intelligent jamming								layer
6.Hello flooding	layer	layer		layer	layer	layer	layer	fake packet injection
7.Flooding								fake packet injection
8.Desynchroni-zation								fake packet injection
9.Sinkhole				Malicious node	Malicious node	Malicious node	routing change	
10.Code injection								
11.Blackhole						layer		
12.Greyhole				Drop packet		layer		
13. Wormhole						layer		
14. Sybil		layer		layer	layer	layer		
15. Sleep privation								

6. Conclusion and future works

The aim of this work was to refine the characterization of attacks in WSN on the basis of relevant aspects. This objective was achieved on sixteen attacks considering all layers of the WSN protocol stack. We investigated and provided UML-based knowledge on the attackers' strategies and the different interactions between malicious entities and legitimate nodes. Furthermore, we classified the attacks according to their similarities and dissimilarities. This work is presented as a clear view and in-depth insight into countermeasures to be designed for WSN. In future works, we will be focusing on extending the number of attacks to be studied by considering the creation of scenarios based on collaboration diagrams to better represent the the relationships that exist between attacks and subsequently a logical representation based on ontological intelligence that can be exploited by a system to enable detection of these attacks.

References

[1] A. Doumi, "La Sécurité des Communications dans Les Réseaux de Capteurs Sans Fils," Ph.D. dissertation, University of Mohamed Boudiaf - M'SILA, wilaya de M'sila, jun 2010.

[2] DataBridgeMarketResearch, "Wireless Sensor Network Market – Global Industry Trends and Forecast to 2028," 2022. [Online]. Available: https://www.databridgemarketresearch.com/reports/global-wireless-sensor-network-market

[3] M. R. Senouci and A. Mellouk, "Wireless Sensor Networks," in *Deploying Wireless Sensor Networks*. Elsevier, 2016, pp. 1–19.

[4] A. P. Singh, A. K. Luhach, X.-Z. Gao, S. Kumar, and D. S. Roy, "Evolution of wireless sensor network design from technology centric to user centric: An architectural perspective," *International Journal of Distributed Sensor Networks*, vol. 16, no. 8, p. 155014772094913, aug 2020.

[5] M. Majid, S. Habib, A. R. Javed, M. Rizwan, G. Srivastava, T. R. Gadekallu, and J. C.-W. Lin, "Applications of Wireless Sensor Networks and Internet of Things Frameworks in the Industry Revolution 4.0: A Systematic Literature Review," *Sensors*, vol. 22, no. 6, p. 2087, mar 2022.

[6] M. Burhan, R. Rehman, B. Khan, and B.-S. Kim, "IoT Elements, Layered Architectures and Security Issues: A Comprehensive Survey," *Sensors*, vol. 18, no. 9, p. 2796, aug 2018.

[7] H. Singh and D. Singh, "Concentric Layered Architecture for Multi-Level Clustering in Large-Scale Wireless Sensor Networks," in *ICSCCC 2018 - 1st International Conference on Secure Cyber Computing and Communications*. Institute of Electrical and Electronics Engineers Inc., jul 2018, pp. 467–471.

[8] S. Abirami, "A Complete Study on the Security Aspects of Wireless Sensor Networks," *Lecture Notes in Networks and Systems*, vol. 55, pp. 223–230, 2019.

[9] I. Butun, P. Osterberg, and H. Song, "Security of the Internet of Things: Vulnerabilities, Attacks, and Countermeasures," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 1, pp. 616–644, jan 2020.

[10] A. Arshad, Z. M. Hanapi, S. Subramaniam, and R. Latip, "A survey of Sybil attack countermeasures in IoT-based wireless sensor networks," *PeerJ Computer Science*, vol. 7, pp. 1–33, 2021.

[11] A. O. Salau, N. Marriwala, and M. Athaee, "Data security in wireless sensor networks: Attacks and countermeasures," in *Lecture Notes in Networks and Systems*, vol. 140. Springer Science and Business Media Deutschland GmbH, 2021, pp. 173–186.

[12] Y. Wu, D. Wei, and J. Feng, "Network attacks detection methods based on deep learning techniques: A survey," 2020.

[13] K. Jane Nithya and K. Shyamala, "A Systematic Review on Various Attack Detection Methods for Wireless Sensor Networks." Springer, Singapore, 2022, pp. 183–204.

[14] S. Hong, S. Lim, and J. Song, "Unified Modeling Language based Analysis of Security Attacks in Wireless Sensor Networks :A Survey," *KSII Transactions on Internet and Information Systems(TIIS)*, vol. 5, no. 4, pp. 805–821, 2011.

[15] S. H. Lee, "A Survey Study on Standard Security Models in Wireless Sensor Networks," *Journal of Convergence Society for SMB*, vol. 4, no. 4, pp. 31–36, 2014.

[16] J. Wang, A. O. Fapojuwo, C. Zhang, and H. Tan, "UML modeling of cross-layer attack in wireless sensor networks," *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, vol. 190, pp. 104–115, 2017.

[17] C. Duru, A. Aniedu, O. T. Innocent, and A. E. E.O, "Modeling of Wireless Sensor Networks Jamming Attack Strategies," *Ameri-can Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, vol. 67, no. 1, pp. 48–65, apr 2020.

[18] M. Amarlingam, P. K. Mishra, K. V. Prasad, and P. Rajalakshmi, "Compressed sensing for different sensors: A real scenario for WSN and IoT," *2016 IEEE 3rd World Forum on Internet of Things, WF-IoT 2016*, pp. 289–294, feb 2017.

[19] D. E. Boubiche, S. Athmani, S. Boubiche, and H. Toral-Cruz, "Cybersecurity Issues in Wireless Sensor Networks: Current Challenges and Solutions," *Wireless Personal Communications*, vol. 117, no. 1, pp. 177–213, mar 2021.

[20] M. V. Pawar and A. Jagadeesan, "Detection of blackhole and wormhole attacks in WSN enabled by optimal feature selection using self-Adaptive multi-verse optimiser with deep learning," *In-ternational Journal of Communication Networks and Distributed Systems*, vol. 26, no. 4, pp. 409–445, 2021.

[21] M. V. Pawar and A. J, "Detection and prevention of black-hole and wormhole attacks in wireless sensor network using optimized LSTM," *International Journal of Pervasive Computing and Communications*, 2021.

[22] N. A. Alrajeh, S. Khan, and B. Shams, "Intrusion Detection Systems in Wireless Sensor Networks: A Review," *International Journal of Distributed Sensor Networks*, vol. 9, no. 5, p. 167575, may 2013.

[23] M. Quentin, "Modèles et mécanismes pour la protection contre les attaques par déni de service dans les réseaux de capteurs sans fil," Ph.D. dissertation, University of Paris-Est, Paris, jul 2015.

[24] L. K. Ramasamy, F. Khan K. P., A. L. Imoize, J. O. Ogbebor, Kadry, and S. Rho, "Blockchain-Based Wireless Sensor Networks

[25] H. Xie, Z. Yan, Z. Yao, and M. Atiquzzaman, "Data collection for security measurement in wireless sensor networks: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2205–2224, apr 2019.

[26] D. Waltenegus and P. Christian, *Fundamentals of Wireless Sensor Networks: Theory and Practice* . Wiley, 2011.

[27] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: Attack and defense strategies," *IEEE Network*, vol. 20, no. 3, pp. 41–47, may 2006.

[28] A. BOUAISSA and M. A. MESSOUS, "Supervision de l'opération d'irrigation des arbres fruitiers à l'aide des réseaux de capteurs sans fil," Ph.D. dissertation, University of Saad Dahlab, Faculty of Science, Blida, jul 2010.

for Malicious Node Detection: A Survey," *IEEE Access*, vol. 9, pp. 128 765–128 785, 2021.

[29] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," in *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, SNPA 2003*. Institute of Electrical and Electronics Engineers Inc., 2003, pp. 113–127.

[30] J. Y. Yu, E. Lee, S. R. Oh, Y. D. Seo, and Y. G. Kim, "A Survey on Security Requirements for WSNs: Focusing on the Characteristics Related to Security," *IEEE Access*, vol. 8, pp. 45 304–45 324, 2020.

[31 d. S. Jessye, "Réseaux de capteurs et vie privée ," Ph.D. dissertation, University of Grenoble Alpes, 2017.

[31] R. Muhammad Noman, B. Attaullah, and M. Athar, "Classification of Attacks on Wireless Sensor Networks: A Sur," *International Journal of Wireless and Microwave Technolog*, vol. 6, pp. 15–39, nov 2018.

[22] H. K. Patil and T. M. Chen, "Wireless Sensor Network Security: The Internet of Things," *Computer and Information Security Handbook*, pp. 317–337, jan 2017.

[32] A. Francillon and C. Castelluccia, "Code injection attacks on harvard-architecture devices," *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 15–25, jan 2009.[Online]. Available: https://arxiv.org/abs/0901.3482v1

[33] C. Hennebert and J. D. Santos, "Security protocols and privacy issues into 6LoWPAN stack: A synthesis," *IEEE Internet of Things Journal*, vol. 1, no. 5, pp. 384–398, oct 2014.

[34 A. Diaz and P. Sanchez, "Simulation of Attacks for Security in Wireless Sensor Network," *Sensors*, vol. 16, no. 11, p. 1932, nov 2016.

[35] M. Shahriar and J. Hossein, "A comparison of link layer attacks on wireless sensor networks ," *International journal on applications of graph theory in wireless ad hoc networks and sensor networks*, vol. 3, no. 1, pp. 35–56, mar 2011.

Tiguiane Yelemou is holder of a doctorate (PhD) in computer science and application from University of Poitiers (France). He is currently a research professor at the dÉcole Supérieure d'Informatique (ESI) in Bobo-Dioulasso. Professor Tiguiane Yélémou's research topics relate to sensor networks, IoT, wireless transmissions and Cybersecurity.

Benjamin Savoudsou received his B.Sc.in the Department of Mathematics and Computer Science in Faculty of Science at the University of Ngaoundéré in Cameroon in 2018. He held his M.Sc. in software and systems in distributed environments in the same institution in 2021. His research interests include Cybersecurity and Wireless Sensor Networks.

Franklin Tchakounté is an Associate Professor and researcher in computer science with more than 10 years of experience in cybersecurity and data science with a strong background in distributed systems. He earned his PhD Degree in Mobile Security from the University of Bremen in 2015. He authored books, book chapters and several research papers in the area of cyber security. He is the founder of Cybersecurity with Computational and Artificial Intelligence (CyComAI) research group. He is fellow of DAAD Staff Exchange in Sub-Saharan-Africa, Research Mobility grants in Ministry of Higher Education in Cameroon, and WebWeWant F.A.S.T project. Devoted to volunteering in reviewing and conference involvement, he has been (senior) member of EAI, ACM, UWB and ISOC SIG. His interests include cyber security and artificial intelligence.

Blaise Omer Yenké is an Associate Professor and researcher in Computer Engineering. He is the Head of Department of Computer Engineering at the University Institute of Technology of the University of Ngaoundéré in Cameroon. He defended his PhD degree in 2010 in an international joint supervision between the University of Yaounde 1 in Cameroon and the University of Grenoble in France. His current research interests include Distributed Systems, High Performance Computing, network modeling, simulation, Sensor Networks Design and Sensor's Architecture.

Marcellin Atemkeng received the Ph.D. with specialization in big data and statistical signal processing from Rhodes University, South Africa, in 2016. He was a Signal Analyst with VASTech, a telecommunication industry in Stellenbosch, South Africa. From 2017 to 2018, he was a Postdoctoral research fellow working with the Square Kilometre Array related projects. Since 2019, he is a Senior Lecturer in the Department of Mathematics at Rhodes University and the coordinator of Rhodes AI Research Group (RAIRG). He has published several articles in peer-reviewed journals and was the 2019 recipient of the Kambule Doctoral Award for recognising and encouraging excellence in research and writing by doctoral candidates at African universities in any area of computational and statistical sciences. His research interests include big data, statistical signal processing, artificial intelligence, deep learning and radio interferometric techniques and technologies.