Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 7.056

IJCSMC, Vol. 10, Issue. 2, February 2021, pg.15 – 30

A THREAT ANALYSIS ALLOWING PROGRESS IN THE SECURITY OF WIRELESS HETEROGENEOUS SENSOR NETWORKS

Nirmi Hajraoui[1]; N. Raissouni[2]; H. Fares[3]
University of Abdelmalek Essaâdi, ENSA-Tetouan, Department of Telecommunication
Laboratory of Remote Sensing and Geographic Information System (LRSGIS)
Mhanech2, Tetouan, 93002, Morocco
[1] hajraouinirmi@gmail.com; [2] naoufal.raissouni.ensa@gmail.com; [3] professeurhajar@gmail.com

DOI : 10.47760/ijcsmc.2021.v10i02.003

ABSTRACT— Wireless heterogeneous sensor networks are of considerable importance. They allow the detection of information coming from inaccessible or hostile areas, the surveillance, control and permanent monitoring of several events in the environment. Because of the irresponsible behavior of some people, the operation of these systems is unfortunately exposed to attacks degrading their performance. A large part of researchers are interested in the security of these systems and are trying to develop solutions that detect and identify these attacks. Some used an intrusion detection scheme, others used rules and key management to detect and identify malicious nodes. In this paper, we present an analysis of the procedures pursued to defend against hackers and a new method of combating attacks. This reinforces the security system of wireless heterogeneous sensor networks.

KEYWORDS— aggregation, authentification, communication, cryptography, encryption, Heterogeneous sensors, Hacker, intrusion, information packet, Network attack, Network security, protocol, revocation, radio canal, routing, Wireless sensor.

I. INTRODUCTION [1-12]

The main security issues in Heterogeneous Wireless Sensor Networks (WHSNs) emerge from properties that make them efficient and attractive, such as resource limitation. Energy is perhaps the strongest constraint on the capabilities of any sensor node. Its energy reserve must be conserved to extend its life span and that of all the components of the related network. Most of the time, the information transmitted is redundant because the sensors are generally and geographically collocated. Most of this energy can therefore be saved by aggregating the data. This requires special care to detect spurious data injection, or erroneous data modification, during aggregation operations at intermediate nodes.

Multi-hop wireless communication is another property of WHSN. In addition to providing easy node deployment, wireless communication has the advantage of providing access to hard-to-reach places such as disastrous and hostile terrain. Unfortunately, the range of "motes" radio communication is limited due to energy considerations. Multi-hop communication is therefore essential for broadcasting data in a WHSN. This introduces many security vulnerabilities at two levels: attack of the construction in maintenance of the roads, and attack of the payload by injection, by modification or by deletion of the packets. In addition, wireless communication introduces further vulnerabilities to the link layer by opening the door to jamming and denial of service-style attacks that lead to battery drain.

The tight coupling with the environment is also a property of WHSN. Most WHSN applications require tight deployment of nodes within or near the phenomena to be monitored. This physical proximity to the environment leads to frequent intentional or accidental compromises of the nodes. As the success of WHSN applications also depends on their low cost, sensor nodes cannot afford tamper-proof physical protection. Therefore, a "well-equipped" adversary can extract cryptographic information from these sensor nodes. As the mission of a WHSN is generally unattended, the potential to attack nodes and retrieve their contents is great. Thus, cryptographic keys and sensitive information should be managed in a way that increases the resistance to capture of nodes.

II. CAUSES of THREATS

The threats of WHSNs are very varied. Their main causes identified so far are as follows:

* Resource limitations can be a threatening cause of intrusion, leading to redundant data injection, signal interference, loss of information, depletion of energy….

* Multi-hop communication favors intrusions producing the loss of routing cards, the deletion or modification of information packets, signal interference….

* The tight coupling of WHSNs with the environment allows attackers to break in and extract the contents of one or more base stations to retrieve cryptographic information.

* etc ..

III. CURRENT PROCEDURES to COMBAT INTRUSIONS

There are four functional blocks of security procedures in WHSNs: key management, routing security, data aggregation security, and radio channel access security.

A. Keys management [13-24]

1) *Keys repartion :*The use of keys provides efficient, secure and stable mechanisms. They allow the management of keys used in cryptographic operations. This key management is an essential service for the security of any communication-based system. Under the constraints of WHSN, designing a key management system is a big challenge. Selecting an appropriate cryptographic solution for these sensor networks is another challenge. The design of these keys has a number of constraints. They derive from the properties of WHSNs. They must be taken into account in the design of any cryptographic key management solution. These properties are as follows:

-limited resources,

-lack of infrastructure,

-the possibility of capturing nodes,

-the possibility of revocation of keys or nodes,

-preference of the symmetrical technique….

In this context, a breakdown of the keys that can be applied in WHSN is proposed below. These sensors are assumed to be head clusters or members of a cluster. The indicated distribution of the keys is explained in three matrices, two lines matrices noted $\mathrm{K_m^{BS}}$, $\mathrm{K_l^{CH}}$ and another square matrix noted K_{nv}^{j}. They are all generated by the base station.

The coefficients of the matrix $\mathrm{K_m^{BS}}$ contain all the symmetrical keys of communication of the base station with any cluster head of the network. They should preferably be stored before deployment in any cluster head.

$$K_m^{BS} = [K_1^{BS}, \quad \dots \quad , K_M^{BS}]$$

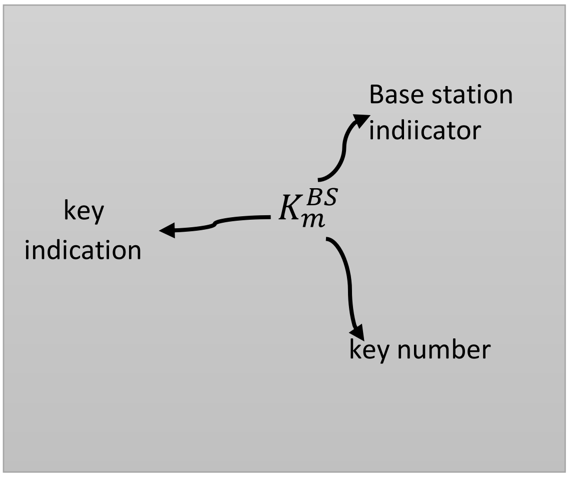


Fig.1: Explanation of matrix symbol K_m^{BS}

The coefficients of the matrix K_m^{CH}, contain all the symmetric keys of communication of any cluster head with each of its members. They should preferably be stored before deployment in any cluster head.

$$K_l^{CH} = [K_1^{CH}, \quad \dots \quad ,K_L^{CH}]$$

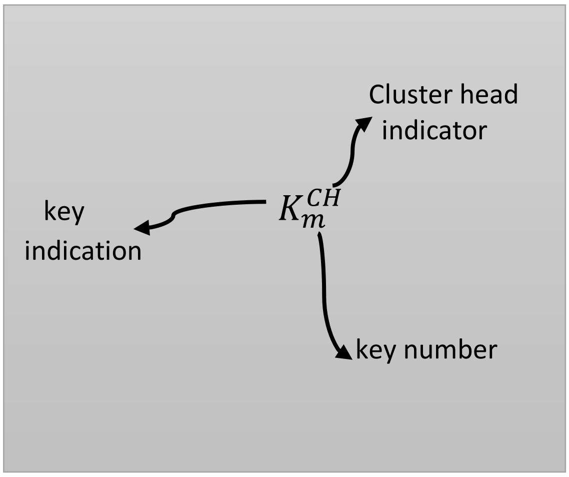


Fig.2: Explanation of matrix symbol K_m^{CH}

The coefficients of the matrix K_{nv}^j contain all the symmetric communication keys of any cluster head. The integer n is number of an origin cluster head. But v is number of a neighbor cluster head. Each symmetric key $(K_{nv}^j = K_{vn}^j)$ is not unique. It can be in J copies, like below:

$$K_{nv}^j = \begin{bmatrix} K_{11}^j & \cdots & K_{1V}^j \\ \vdots & \ddots & \vdots \\ K_{N1}^j & \cdots & K_{NV}^j \end{bmatrix} \quad \text{with}$$

$$K_{nv}^j = [K_{nv}^1, \quad \cdots \quad ,K_{nv}^J]$$

Each row of the K_{nv}^j matrix contains all of the keys that we can have in a single cluster head and that we must load in this same cluster head before deployment. The use of a key to secure a link between base station and cluster head, between a cluster head and any corresponding member or between two head clusters, can be random. That is, the choice of a coefficient in any row of the indicated matrices can be random.

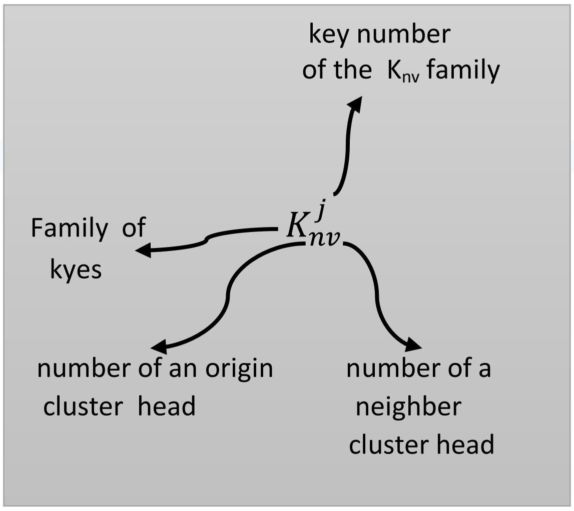


Figure.3: Explanation of matrix symbol K_{nv}^{j}

2) Cryptography processes: Key management is one of the most difficult aspects of setting up a cryptographic security system. For such a system to function and be secure, each user must have a set of secret keys or a pair of public or private keys. This allows to:
* generate keys or offer these users the means to generate them,
* distribute these keys securely to users and be able to exchange them,
* save and store these public and private keys in a secure manner,
* verify the authenticity of all these keys.
In public key systems, management includes the ability to verify and manage the public keys of other users. They are signed in the form of digital certificates.
After deployment, sensors need to establish cryptographic keys with their neighbors to provide security services such as:
• Secure the routing or construction of roads
• Secure data aggregation
• Cooperate in authentication or access to the radio channel, etc

3) Asymmetrical or symmetrical key systems : In public key systems, key exchange is greatly simplified. Each communicating party publishes its public key. Public keys are usually distributed using digital certificates, used by the recipient to authenticate the received public key. All communications with this party will then be encrypted with this key. The main advantage of using public key algorithms is the ease of key management and their reliability. The disadvantages of this approach include the energy consumption due to the calculation of public key algorithms, to the transmission of certificates, and to storage of keys. These public keys are larger in number and size than symmetric keys. Employing symmetric key mechanisms to establish trust dramatically reduces the power consumption of sensor nodes and the storage space reserved to accommodate these keys. However, the exchange of keys in symmetric key systems is much more complicated. Usually, a single symmetric key is used between two communicating parties, over a single session or over a limited period. Although public key cryptography has definite advantages over symmetric key cryptography and despite research aimed at applying them to WHSN, symmetric key cryptography has its own qualities which still make it the most preferred for the indicated network. For this reason, most of the key management methods proposed for WHSN are based on symmetric cryptography. The major problem with symmetric cryptography is being able to find a method that makes it easier to establish keys between nodes. The common solution is to use a pre-distribution method, in which the keys are loaded into the sensor nodes before deployment. The coordination of pre-distribution-based key management solutions is linked to protocols. In this repartition, these protocols are agenced according to the way in which neighboring nodes share common keys, probabilistic or deterministic, and according to a hierarchical or flat network topology.

4) *Random pre-distribution of keys :* In the past, researchers[19] have proposed a key management method, based on the probability of sharing a key between the nodes of a random graph. It provides techniques for: key pre-distribution, shared key discovery, key path establishment, and key revocation. The main idea of this method is to randomly distribute a certain number of keys, from a finite set, to each node of the network before its deployment. This way, any two nodes will only be able to exchange secure messages if they have a common key. This pre-distribution process comprises three phases: Key pre-distribution phase: In general, a large set S of keys is generated at the start. For each node, m keys are chosen at random from the set S:

$$S= \{(K_{id1}= key_1), (K_{id2}= key_2),....\} \text{ with}$$

$$id_j : \text{identifier number j} \quad \text{and} \quad Key_j : \text{key number j}$$

These m keys are stored in the memory of the node and form the keyring of this node. The number of keys in the set is chosen such that two random subsets of S of size m will have a certain probability p of having at least one key in common. This common key is identified by execution of the following two phases:

*****Discovery phase of shared keys

Nodes discover their neighbors and more particularly those with whom they are able to communicate securely. Because, they have an identical key in their respective keyrings. The pre-distribution protocol may be to broadcast the list of keys owned by. The key shared between two nodes becomes the session key of their link.

*****Path establishment phase based on existing keys: After the shared key discovery phase, the network becomes a connected graph made up of a few secure links. The network nodes can then use the existing links to set up shared keys with their neighbors who did not share a common key with them.

5) *Revocation of keys:*The revocation of a compromised node, identifier id, is done by eliminating its keyring. To do this, a controller node which has great connectivity announces a simple revocation message. This contains a signed list of K_{dj}^{i} keys to be removed from the keyrings of other nodes. The list of keys to be withdrawn is signed by another signature key K_e generated by the base station BS and sent by unicast to each node i by encrypting it with the key K_i^{BS}. The key K_i^{BS} is shared between the base station and the ith node during the key pre-distribution phase. Some links will be lost due to the deletion of keys from the compromised node. This requires a reconfiguration of these links by the discovery of shared keys or the establishment of secure paths.

6) *Multiplication of symmetrical keys :* According to the results in the paper designated in the reference [21], the process is identical to that indicatred in the above section, except that instead of requiring the sharing of a common key to secure a link, a pair of nodes must share q keys with q>1 to establish a secure link. The new key used for communication between these two nodes is the hash of all shared keys [14,15]. For example, for any two nodes sharing q' keys (q'≥q), the key used for communication, is used for communication between these two nodes is the hash of all shared keys. For example, for any two nodes sharing q' keys (q'≥q), the key used for communication, is

$$K = hash\ (K_1 \parallel K_2 \parallel ... \parallel K_{Q'}).$$

The more the number of shared keys increases, the more the resilience against capture of the node increases. Otherwise, as the required number of shared keys increases, it becomes more difficult for an attacker with a given set of keys to break a link. However, to preserve a given probability p for two nodes to share sufficient keys and establish a secure link, it is necessary to reduce the size of the set of S keys. This must be, even if this allows an attacker to win a plus K_{id}^{j} large number of samples of S to break knots.

7) *Extendable and lightweight authentication protocol(leap):* This is a deterministic key management protocol for wireless sensor networks. The key management mechanism provides, through LEAP support [23], the internal "in-network processing" while limiting the security impact of a compromised node on its immediate neighborhood in the network. LEAP supports the establishment of four types of keys for each node: global key, individual key, pair key and group key.

**Operating hypothesis: LEAP is based on a transient initial key K_{IN} loaded in each of the network nodes. The authors of LEAP assume that to compromise a node, the adversary requires a minimum time T_{min}. Now is the time to plug in a serial cable and copy the contents of the compromised node's memory. LEAP exploits this trust time to allow two neighboring nodes to establish, in a secure manner, a symmetric session key from the transient initial key K_{IN}. After this time T_{min}, the K_{IN} key is deleted from the memory of the node. We therefore proceed according to the following steps:
*****Loading initial key: The BS generates an initial key K_{IN} and loads each node with this key. Each node u generates a master key $K_u=f_{KIN}(u)$, f_{KIN} being a pseudo-random function.
*****Discovery of neighbors: Immediately after its deployment, node u tries to discover its neighbors by broadcasting a HELLO message which contains its id. Also, it initiates a timer which will be triggered after the time T_{min}. Node u expects an ACK from each of its neighbors v containing the identifier of v. The ACK is authenticated using the master key K_v, which is generated as follows: $Kv=f_{KIN}(v)$. Since node u has the K_{IN} key, it will also be able to generate K_v, so it can verify the authenticity of the ACK received.
*****Establishment of the pairwise key: The node u calculates its pairwise key K_{uv} with v, as follows: $K_{uv}= f_{Kv}(u)$. The node v can also calculate K_{uv} in the same way. K_{uv} serves as a key between nodes u and v. It is a symmetric key.
*****Deletion of the keys: When the timer expires after T_{min}, the node u deletes K_{IN} and all the main keys K_v of its neighbors. Note that node u does not erase its main key K_u.
****** LEAP security: At the end of these four steps, the node u will have established a key per pair K_{uv} shared with each of its neighbors. This key will be used to secure the data exchanged between them. In addition, no node in the network has the K_{IN}. An adversary can eavesdrop on all traffic in this phase, but without the $K_{IN.}$ They cannot inject incorrect information or decrypt messages. An adversary compromising a node after T_{min}, obtains only the keys of the compromised node. When a compromised node is detected, its neighbors simply delete the keys that have been shared with it.

B.Routing safety in the whsn[24-28]
In Intrusion-tolerant routing for WHSN, the routing layer is the module responsible for correctly routing data from one point in the network to another. This layer is made up of two functional blocks: one for building roads and the other for relaying data. The first component makes it possible to build a backbone connecting the nodes to the desired destinations via a set of paths. The second component uses this backbone to route the collected data to end users. An adversary wishing to attack the network can then attack one of the two components that must be protected.
In the attacks of the routing protocol, most protocols are quite simple, and therefore quite vulnerable to attacks because the constraints of WHSN are imposed. A malicious node can operate in two situations:
• In the case of data exchanged between the nodes,
• In the case of a network topology created by a protocol.
These threats can be divided into two categories: active and passive attacks.

1) Active attacks in routing:To defend yourself against any attacker, we must know his procedure or his philosophy. These attacks are very varied. Each one of them, is useful in firewall conception or in security at all. Herewith are some examples:

+Jamming attack: Due to the sensitivity of wireless media to noise, a node can cause a denial of service by emitting signals at a certain frequency. This attack can be very dangerous. Because it can be carried out by an unauthenticated person foreign to the network. This intruder can send a lot of packets. Relaying these packets, depletes the energy of the sensors and / or will jam the useful signal.

+Sinkhole attack: In a sinkhole attack, the node tries to attract as many paths as possible to it allowing control over most of the data circulating in the network. In order to do this, the attacker must appear to the other nodes as being very attractive, presenting optimal routes. Having become an important router by mistake, it begins to drop packets or even all packets.

+Wormhole attack :In a wormhole attack, an attacker receives packets at a point on the network, then encapsulates them to another attacker to reintroduce them into the network.

Encapsulation can be done in two ways. In multi-hops, this encapsulation makes it possible to hide the nodes located between two attackers. Therefore, the paths passing through the malicious node appear shorter. This makes it easier to create sinkholes with protocols that use the number of hops as a path choice metric. In direct communication, routes passing through attackers are faster. Because, they are a jump away. Therefore, this technique can be used against protocols that rely on route latency or on those that use the first route discovered.

+Routing table poisoning: Some optimizations have been developed in order to increase knowledge of paths. When a node hears (in promiscuous mode) routing information, it updates its local routing table accordingly. A malicious node can send out a large number of false information, thus filling the node routing tables. As these tables have limited sizes, this will generate an overflow, and the tables will only contain false routes.

+Attack sybil: In some algorithms, the reliability of the routing is implemented by the establishment of a redundancy of paths. An attacker can alter this kind of systems by endorsing several identities, which allows to create several routes passing through the malicious node, which are in reality only one path.

+Hello flooding attack: The short range of sensors and the presence of a laptop class attacker allowed the introduction of a new attack called hello flooding. This attack is based on the fact that most of the links between the laptop attacker and the sensors are unidirectional. Therefore, an attacker can broadcast the information of an optimal route to all the nodes of the network by transmitting with a strong signal. All nodes will update their local tables. When a node wants to communicate, it will not be able to use this route because the next hop, which is the attacker, is out of range.

2) *Passive attacks in routing :* **-**Lack of cooperation or selective forwarding attack: all routing protocols assume that nodes are "honest" and will normally relay packets that pass between them. However, an attacker can violate this rule by dropping all or part of these packets. Additionally, if the attacker has previously used a sinkhole attack, he becomes an important router in the network. So, by giving up its role of router, the performance of the system will be seriously degraded.

-Eavesdropping attack: Since wireless media is open media, a node can hear all communications from its neighbors.

This can reveal important information, such as the location of an important node. The combination with a sinkhole attack further worsens the impact of this attack.

3) *Defense procedures against routing attacks:* We distinguish three types of protection against attacks on data routing in WHSN:

Prevention solution against active attacks: in this category, cryptographic mechanisms are generally used to protect the signage used for road construction. They are generally authentication and integrity control mechanisms that are used to prevent a malicious node from injecting, modifying and / or deleting information that will be used for the discovery, construction or maintenance of a road.

Solution for detecting suspicious behavior: in this category we try to detect behaviors that testify to a passive attack such as lack of cooperation, refusal of packet relay, etc ...

Tolerance solution for the omission of the nodes due to attacks: in this category, mechanisms are introduced for tolerance of failure of nodes due to attacks or failures. Multi-path routing is a typical example.

Therfore, In intrusion-tolerant routing for wireless sensor networks protocol, the idea is to allow the SB to draw a correct mapping of the network which will make it possible to establish the routing tables for each sensor. These tables will then be transmitted to the nodes concerned in a secure manner. This protocol has the below two objectives :

>Operate correctly in the presence of intruders: The indicated protocol is an intrusion-tolerant one, which ensures routing even in the presence of intruders. It builds several independent paths for each pair of communicators. Independent paths only share a small number of nodes / links. But, they share the source and the destination. With this property, an intruder will only be able to alter at a given time, communications crossing a single path.

>S<u>calability and energy saving</u>: The calculation of independent paths and the establishment of routing tables represent a rather heavy task. The protocol indicated above, performs these calculations in the SB. The results are then transmitted to each node in a secure manner

>A<u>uthenticated initiation of the construction of the tree</u>: The SB must first draw up a tree covering the entire network, of which it is the root. This tree will allow the routing of control messages between the sensors and the SB. In order to avoid SB spoofing, the above protocol employs an authenticated broadcast mechanism. To do this, the SB generates a one-way hash string (n_i) $0 \leq i \leq k$ as follows:

$$n_{i+1}=h(n_i),\ 0<i<k$$

where no is chosen randomly, nk is known by all nodes and h is a one-way hash function. Periodically, the SB generates a request to reconstruct the routing tables of the sensors. The format of the message is as follow:

$$type|ows|size|path|MACR_x$$

The ows (One-Way Sequence number) field designates the current value of the hash string (i.e. for the request i, ows = n_i). Each node locally maintains the last value received from ows, referred to as owsfresh. When a node x receives a request, it checks its freshness and authenticity by the following relation:

Find j, such that j> 0 and

$$ows=h^j(ows_{fresh})$$

>C<u>onstruction of link tree by relaying the request</u>: a node x which receives the previous request message, adds its identifier to the path field and calculates the MACRx field such as: MACRx = MAC (Kx, size | path | ows | type) where Kx represents the secret key shared between node x and the SB. This node must also send its "upstream" to the SB, which represents its first neighbor which sent a valid request and saves its MACR. It will be referred to as $MACR_{upstream.}$.

>R<u>oute feedback</u>: After issuing the request, the corresponding node waits a certain time to send feedback to the BS. This period allows him to collect information on his neighborhood. This information will allow the BS to establish a global view of the network in a secure manner. A feedback message contains the following information:

$$type|ows|path\_info|nbr\_info|MACR_{upstream}|MACF_x$$

The "path_info" field contains the list of nodes between node x and BS on the one hand and between the identification of x and its MACR on the other. The "nbr_info" field contains the list of neighbor identifiers as well as their MACRs. The MACFx field is calculated as follows:

$$Id_x|size|path|MACR_x$$
$$size|Id_a|MACR_a|Id_b|MACR_b|\ldots$$
$$MACF_x=MAC(k_x,\ path\_info|nbr\_info|ows|type)$$

To reinforce security, the message is routed using the $MACR_{upstream}$ field. Thus, when a node receives a feedback message, it compares the value of the $MACR_{upstream}$ received with its value linked to the ows received, and possibly relays the message.

>C<u>onstruction of routing tables</u>: After issuing the request, the BS waits a certain period before starting the construction of the routing tables. During this period, it processes the feedback messages received in order to draw up the network graph. To check the feedback information of a node x, the BS recalculates the MACFx. It must also compare the MACR values of each neighbor v with the MACRv value received in the feedback from node i . Thus, the falsified information is rejected, and a correct graph can be established. The BS can then search for the independent paths and construct the relay tables for each node accordingly. The independent paths are chosen so that they group together the fewest possible nodes in common. The algorithm used employs a simple method, which first searches for the shortest path between each pair of communicators. Subsequently, the

algorithm tries to find another path in the subgraph that does not contain the nodes of the first path, their neighbors and the neighbors of the neighbors. If this is impossible, the process is reiterated by adding all the neighbors of the neighbors. Then in case of failure, we add all the neighbors. For each sensor, the BS calculates a relay table that contains an entry for each path passing through the sensor. This table is encapsulated in the following message:

type|ows|size|routingTable|MAC
où le MAC est calculé comme suit :
MAC=MAC(k_X,type|ows|size|table)

4) *Secroute protocol :* This tool is a secure hierarchical routing protocol. The network is organized into clusters each having a leader. The collector node is supposed to know this organization of the network, and must locally maintain a table containing a secret key for each sensor. This key is supposed to be pre-loaded in each node. In addition, each cluster must have a key to secure intra-cluster exchanges. This key must be known by the cluster-head and all the nodes of its group. The SecRoute protocol does not specify the cluster construction algorithm, and assumes that the clusters and their keys are established by another protocol, such as LEAP [23].

=Properties of secroute : This protocol has the following properties [27]:
* Routing packets are not large. Because they only contain partial information on the path traveled.
* This protocol uses a two-tier architecture, where leaders aggregate member data and then pass it to the collector node.
*This protocol only employs symmetric encryption methods.
* For security reasons, the SecRoute protocol replaces unicasts with local targeted broadcasts. Indeed, a message sent while avoiding unicasts is received by all the neighbors. This makes it possible to check, during the relay, the integrity of the message sent by the next hop. Each sensor stores a routing table with the following format:

Source	Pre	Next
Id_{Source}	Id_{pre2} , Id_{pre1}	Id_{next1} , Id_{next2}
⋮	⋮	⋮

Table.1 :Format of the routing table in SecRoute

The table is organized according to the address of the message source. The Pre (respectively Next) field indicates the next two hops to the BS (respectively to the source) on the path between the source and the BS.

=Path discovery: The source node initiates a path discovery by sending to its direct neighbors RREQ a request packet containing the following information:

$Id_{source}|Id_{sink}|Id_{RREQ}|N_{source}$ MAC(K_{source}, $Id_{source}|Id_{sink}|Id_{RREQ}|N_{source}$) where

* Id_{source}, Id_{sink}, Id_{RREQ}: are the identifiers of the source, the collector and the request respectively.

* N_{source}: a nonce generated by the source. When a node receives a request, it is only accepted with the uniqueness of its identifier Id_{OO}. It then updates its routing table using the information from the two previous hops to the source. Before relaying the request, the node replaces the values of Id_{pre} and Id_{this} with Id_{this} and its identifier respectively.

$Id_{this}|Id_{pre}|Id_{source}|Id_{sink}|Id_{RREQ}|N_{source}$ MAC (K_{source},
$Id_{source}|Id_{sink}|Id_{RREQ}|N_{source}$)

=Response relay: When the collector receives the first request, it checks the MAC constructed by the source using the key relating to its identifier Id_{source}, saved in the local table. If the MAC is correct, the collector updates its routing table using the Id_{pre} and Id_{this} fields. It then generates an RREP response having the following format:

$$Id_{pre}|Id_{this}|Id_{next}|Id_{sink}|Id_{RREQ}|N_{sink}\ MAC\ (K_{source},\ Id_{source}|Id_{sink}|Id_{RREQ}|N_{sink})$$

The request is sent in local broadcaste, targeted using the Id_{next} field. When the neighbor sensor having the Id_{next} identifier receives this response, it updates its routing table accordingly, then replaces the Id_{pre} and Id_{this} fields with Id_{this} and its identifier. It must also modify the Id_{next} field by the known identifier during the path discovery phase. If the node does not receive the response from Id_{next} after a certain time, it ignores all requests made during the next discovery phase. The relay node must also check that the response sent by the next hop is valid, by ensuring that it is indeed intended for the two-hop node contained in the path to the source (ie the Id_{pre2} field of the routing table). When the source receives the first response, it checks the MAC generated by the collector and updates its routing table adding Id_{this} and Idpre as next hops to the collector.

=Data relay: Data relay is performed in two stages. Member nodes send their data to the group leader, who will then send the summary to the collector. The establishment of the path between the leader and the collector is carried out thanks to the process described previously, i.e.: the leader represents the source of his group. For intra-group communications, the protocol uses the CK cluster key established when it was created. Each data D of a sensor is sent to the clusterhead Id, encapsulated as follows:

$$Id|\{D\}CK|MAC(CK,\ Id|\{D\}CK)$$

The leader aggregates the member data after checking the MACs of each packet, then sends the summary to the collector. For this, the manager operates as the source of the data. If no route to the collector is known, the path discovery process should be performed. Using the constructed path, the leader emits the data in the following packet:

$$Id_{this}|Id_{next}|Id_{source}|Id_{sink}|Q_{ID}|\{D\}K_{source}\ MAC\ (K_{source},\ Id_{source}|Id_{sink}|Q_{ID}|\{D\}K_{source})$$

An intermediate node with the same identifier as Id_{next} relays the packet, replacing Id_{this} and Id_{next}. If a relay node does not receive the packet transmitted by the next hop, route maintenance must be performed. To do this, it must send an error message to the source allowing the route discovery procedure to be started again. In addition, the next hop is added to the black list. This list contains the identifiers which the node should ignore their response packets. When the packet reaches the collector, it checks the MAC using the source key and can therefore decrypt the data and use it.

C.Security of aggregation in WHSN [29-34]

1) *Attacks on data aggregation :* Data aggregation may be attacked. It is necessary in the WHSN to minimize redundant transmissions and therefore save energy. In order to perform a secure aggregation operation, an intermediate node must have access to the data transmitted by its pairs. This is to calculate the useful information by using an aggregation function like the average, the maximum, the minimum etc… A malicious node can then attack these operations by injecting false data into the network or by falsifying the result of an aggregation operation. In this case, the malicious node will succeed in falsifying the information captured in an entire network area. The risk can still be incurred by a bad aggregation operation. The whole difficulty is how to allow the relay nodes to have access to the intermediate results allowing to calculate the useful information by aggregation and to make this operation free from the risk of falsification, deletion or modification?

2) *Different of secure aggregation solutions* : There are two main categories of solutions depending on the cryptographic mechanism used:
*The first solution is based on end-to-end encryption, taking into account the flat or hierarchical network topology. In this category, cryptographic mechanisms are used which secure the information captured from end to end, while allowing intermediate nodes to perform aggregation operations. The information here is generally only checked at the level of the collector, which generates strong contamination by false information.
*The second solution is based on step by step encryption. In this case, the veracity of the information is checked step by step. Its rejection can be done at any level of the tree covering the network.

3) *Secure Aggregation for Wireless Networks (SAWN) :* This protocol assumes that two consecutive nodes cannot be compromised simultaneously. It is based on the two-hop verification. Any node checks whether the aggregation of the data of its grandsons, carried out by its child, is correct. The verification of the aggregation is done in a deferred manner in time using the protocol µTESLA (micro Timed, Efficient, Streaming, Loss-tolerant Authentication)[42]. This is for the authentication of the keys used in the authentication of data and their aggregations. In the following, we assume that the nodes have the means to verify the authenticity of the keys shared between the nodes and the BS, when the latter reveals the keys for verification.

Each leaf node transmits its reading to its father. The messages include the reading of the data of the node, its id, as well as a MAC calculated thanks to the key K_{Ai}. The latter is shared between node A and the base station, but the other sensors do not know it.** The parent node stores the message as well as its MAC until the key K_{Ai} is revealed by the base station. At this moment, it will check the MAC and send an alarm in case of difference. Aggregation of reads is performed in each intermediate step. The nodes wait for a specified time to receive messages from their children and then retransmit the messages and MACs. They receive directly from their immediate children. The nodes aggregate the data they receive from their grandsons (through their children) and pass the MAC of the aggregate value. After all messages arrive at the base station, the base station reveals the temporary keys of the nodes. Once the key K_{Ai} is revealed, the nodes move to the next temporary key K_{Ai+1}.

>>Secure aggregation in sawn: Consider the aggregation tree illustrated by the following figure**.** The example illustrates the i-th round where it uses the K_{xi} keys to authenticate the messages transmitted from one node to another:

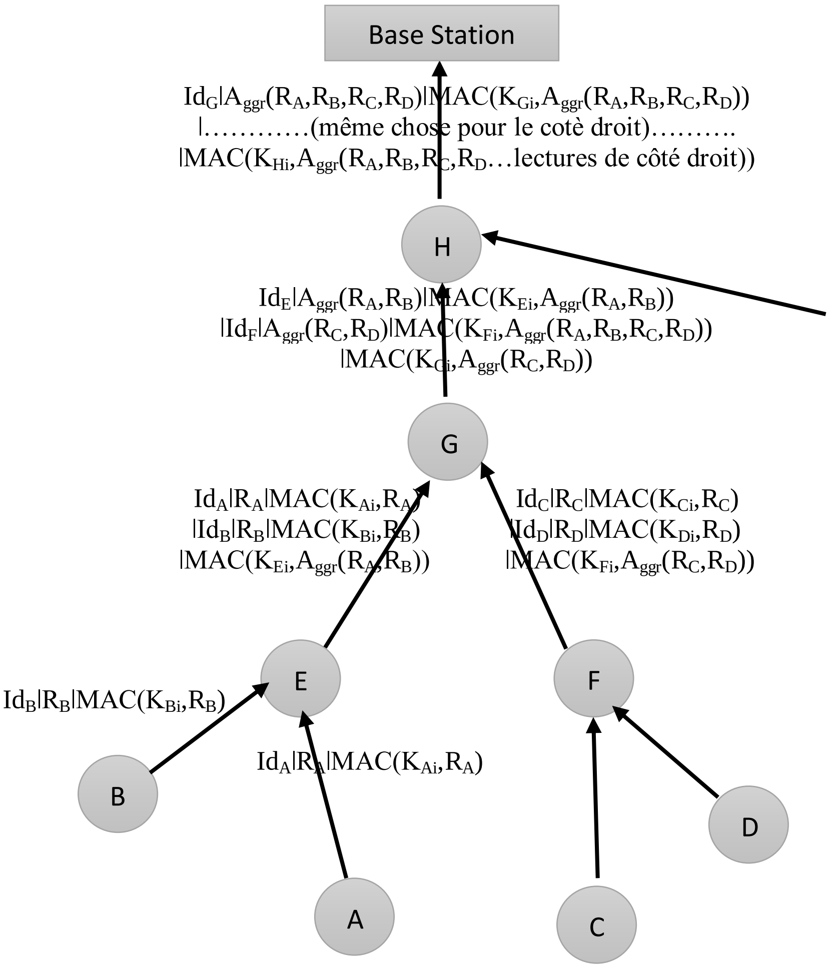

Base Station
$Id_G|A_{ggr}(R_A,R_B,R_C,R_D)|MAC(K_{Gi},A_{ggr}(R_A,R_B,R_C,R_D))$
|…………(même chose pour le cotè droit)……….
$|MAC(K_{Hi},A_{ggr}(R_A,R_B,R_C,R_D$…lectures de côté droit))
H
$Id_E|A_{ggr}(R_A,R_B)|MAC(K_{Ei},A_{ggr}(R_A,R_B))$
$|Id_F|A_{ggr}(R_C,R_D)|MAC(K_{Fi},A_{ggr}(R_A,R_B,R_C,R_D))$
$|MAC(K_{Gi},A_{ggr}(R_C,R_D))$
G
$Id_A|R_A|MAC(K_{Ai},R_A)$
$|Id_B|R_B|MAC(K_{Bi},R_B)$
$|MAC(K_{Ei},A_{ggr}(R_A,R_B))$
$Id_C|R_C|MAC(K_{Ci},R_C)$
$|Id_D|R_D|MAC(K_{Di},R_D)$
$|MAC(K_{Fi},A_{ggr}(R_C,R_D))$
E
F
$Id_B|R_B|MAC(K_{Bi},R_B)$
$Id_A|R_A|MAC(K_{Ai},R_A)$
B
D
A
C

Figure.5: Example of secure aggregation tree with SAWN protocol

>>Data transmission: Nodes A, B, C and D send data to the base station via the aggregation tree built with a routing protocol.
* Leaf nodes send data to their father. The messages include MACs calculated with the current authentication key:

A==>E : R_A | Id_A | MAC(K_{Ai},R_A)

Each key is used to authenticate a single message, which will prevent the replay attack.
* Intermediate nodes receive messages from their children. The parent node cannot yet verify the MAC because the child key will only be revealed during the verification phase. For the moment, the father stores the message and the MAC. The intermediate node waits for packets from the children, and then sends a message to its father containing the reads of the children, their MACs, as well as the MAC calculated on the aggregation value:

E==>G : R_A | Id_A | MAC(K_{Ai},R_A) | R_B | Id_B | MAC(K_{Bi},R_B) | MAC(K_{Ei},Aggr(R_A,R_B))

There is no need to pass the calculated aggregation value, since G can calculate Aggr (R_A, R_B) from the R_A and R_B values. It is also not necessary to pass the Id_E to G, because G knows the topology of the network so can determine which node is sending the message.

*Node G receives messages from nodes E and F. For each of them, G calculates the values of the aggregation of the readings of its grandsons that is to say (A, B, C and D). It then passes the aggregate values of its grandsons, the Id of its children, and their MAC values. G also calculates and transmits the MAC of the following aggregation value:

$$Aggr(R_A,R_B,R_C,R_D)=Aggr(Aggr(R_A,R_B),Aggr(R_C,R_D)).$$

*Since the function of aggregation is known to all nodes, the MAC calculated by E will authenticate the value calculated by G. The sensor readings and MAC values received from E and F are stored for later verification.

$$G ==> H : Id_E \mid Aggr(R_A,R_B) \mid MAC(K_{Ei}, Aggr(R_A,R_B)) \mid Id_F \mid Aggr(R_C,R_D) \mid MAC(K_{Fi},Aggr(R_C,R_D)) \mid MAC(K_{Gi},Aggr(R_A,R_B,R_C,R_D))$$

Likewise, node H receives messages from G and another branch, and in turn transmits the aggregated message to the base station. Note that the length of the message does not increase if the network was deeper.

*The base station receives the message from H. It can calculate the value of the final aggregation, Aggr (R_A, R_B, R_C, R_D, ...) using Aggr (R_A, R_B, R_C, R_D) and the other values of its child nodes.

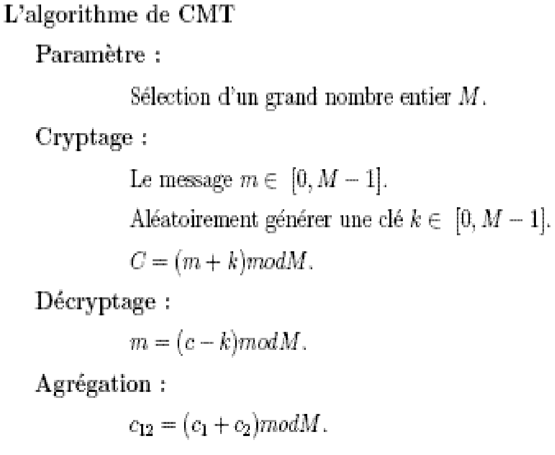
>>Data validation: The purpose of the SAWN protocol is to authenticate all reads that participated in the aggregation value, without receiving all of those reads. To validate the data (reading nodes and aggregation values), the base station reveals the current key K_{xi} that it shares with each node x of the network, by sending a single message containing all these keys. This key disclosure message is authenticated by a MAC using a key authenticated by µTESLA [42]. If a node detects an erroneous message in the data validation step, it sends an alarm message. An alarm is issued by a parent when it detects that the aggregation MAC of a child is inconsistent with the data of the grandsons, or when the MACs of the data itself are in error.

4) *Protocols based on end-to-end encryption:* These protocols in a category which uses a shared key between each node and the collector node to guarantee the integrity of the data transmitted in the network. As the data contents are encrypted, the nodes use a particular type of cryptography called Privacy Homomorphism to be able to perform the aggregation.

=>Privacy homomorphism (PH): Any algorithm may be PH [29] if and only if by having E(x) and E(y), we can calculate E(x¤y) without decrypting x and y. So it checks the following property:

$$E_{K1}(x_1) ¤ E_{K2}(x_2)=E_{K1+K2}(x_1 ¤ x_2),$$

where Ki are the keys and xi are the data. The single point of verification in this type of protocols is the sink node. The latter having all the keys used to encrypt data in the network.
=>CMT protocol is a protocol proposed by Castelluccia, Myldetun and Tsudik. CMT is based on the assumption that each node uses a symmetric key shared between this node and the collector node. The idea of this protocol is that each node makes the modular addition between its stored key and its ring. During the data routing phase, the aggregation is done on this data which is already encrypted. The following algorithm shows the different steps of this protocol:

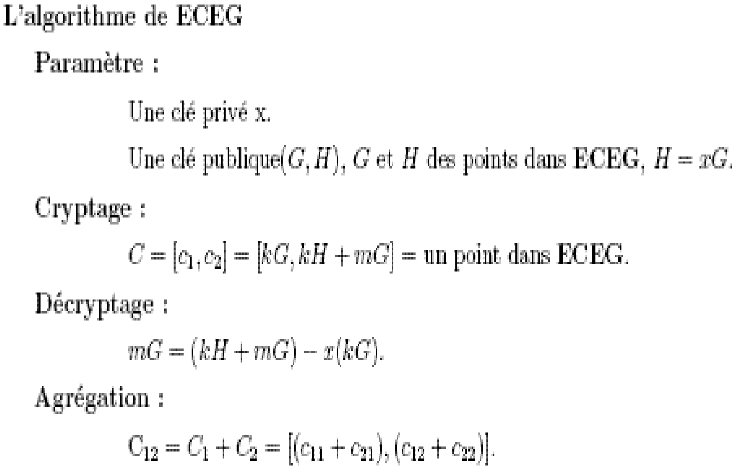


```
L'algorithme de CMT
    Paramètre :
            Sélection d'un grand nombre entier $M$.
    Cryptage :
            Le message $m \in [0, M-1]$.
            Aléatoirement générer une clé $k \in [0, M-1]$.
            $C = (m+k) mod M$.
    Décryptage :
            $m = (c-k) mod M$.
    Agrégation :
            $c_{12} = (c_1 + c_2) mod M$.
```

Algorithme CMT

The size of the packet in this protocol depends on the size of the integer number M, and a single modular addition is sufficient for aggregation and encryption. Thus, this protocol does not consume a lot of energy.

=>Elliptic curve elgamel protocol: This protocol uses an ElGamel Elliptical Curve Cryptographic Algorithm (ECEG) which is an asymmetric approach that does not consume as much power as conventional asymmetric systems like RSA (Rivest–Shamir–Adleman) protocol [11]. The following algorithm illustrates aggregation and verification in this protocol:



L'algorithme de ECEG

Paramètre :

Une clé privé x.

Une clé publique(G,H), G et H des points dans ECEG, $H = xG$.

Cryptage :

$C = [c_1, c_2] = [kG, kH + mG]$ = un point dans ECEG.

Décryptage :

$mG = (kH + mG) - x(kG)$.

Agrégation :

$C_{12} = C_1 + C_2 = [(c_{11} + c_{21}), (c_{12} + c_{22})]$.

Algorithme ECEG

D. Radio channel access security [35-49]

A question always imposed is that of the assumptions under which, the security promises of the physical layer of any wireless network can be kept. A critical aspect here is the level of knowledge of the inherently uncertain wireless environment required to be able to guarantee a certain security performance. A general observation is that the rigor of the notion of physical layer security places more emphasis on the design assumptions of the physical layer and the adequacy of secure transmission strategies. Guided by this information, researchers noted in the reference [49] proposed a probabilistic characterization of secrecy that captures the uncertainty in the communication model arising from information about the unknown channel state of the listening channel. The use of any-to-any radio cards is an integral part of this concept as the underlying contextual information for the probabilistic characterization of the secret. Radio maps are learned using recent advances in machine learning in network channel gain mapping. The resulting privacy maps provide the key ingredient in integrating physical layer security into radio access and represent a valuable tool in the design of secure radio access systems. The statistical characterization focuses on the concept of spatial availability of services in wireless networks, indicating the locations in a given area where the performance related to secrecy would exceed a given threshold with a guaranteed level of confidence. In this sense, it provides a notion of quality of service for security in a spatial context, of the physical layer of any WHSN.

IV. PROPOSITION

Any intrusion or attack detection system need more performance. Many researchers only have focused on the below indicated procedures. We propose in this paper another means of fighting against intrusions into WHSN and against their attacks. The traffic in these networks is carried out with the transmission in one direction or in another of the information packets. Each packet contains different information fields: control, data, synchronization etc …The location of these information fields in each packet has so far been fixed. This location can be changed multiple times in any communication in the WHSN. It would therefore be very interesting to add a code in each packet which indicates the applied rearrangement of fields in any information packet. Changing this rearrangement from one packet to another, or from one group of packets to another group, certainly helps defend against hackers. This procedure requires little or no treatment. Therefore, it definitely reduces power consumption and extends the network life.

V. CONCLUSION

The services rendered by WHSNs are always useful and sometimes essential to the well-being of people. We must constantly ensure their operational safety and contribute without hesitation in this mission. This is what we did in this work, like most other researchers. They still face security challenges taking into account available resources and the mobility of WHSN sensor nodes. Indeed, we have presented here an analysis of the procedures pursued so far in security and we have proposed a design method of new procedures to fight against intrusions and attacks of WHSN. Future work may be focused on analyzing the causes of vulnerability of these networks and the causes of their exposure to attacks. We will think about developing new solutions such as reducing computing power, reducing high network overload, increasing the rate of intrusion detection, increasing circuis integration and applying or validating these concepts in real heterogeneous sensor networks.

REFERENCES

[1] Y.wang, G.Attebury and B.Ramamurty, "A survey of security issues in wireless sensor networks", in IEEE Communication Survey Tutorials, 2006.

[2] M. Bloch, J. Barros, M. R. D. Rodrigues and S. W. McLaughlin, "Wireless information-theoretic security", IEEE Transactions on Information Theory, vol. 54, no. 6, pp. 2515-2534, June 2008.

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

[4] M. Bellare, S. Tessaro and A. Vardy, "Semantic security for the wiretap channel" in Advances in Cryptology - CRYPTO 2012, Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 294-311, 2012.

[5] J. H. Lee, L. H. Kim, and T. Y. Kwon, “FlexiCast: energy-efficient software integrity checks to build secure industrial wireless active sensor networks,” IEEE Transactions on Industrial Informatics, vol. 12, no. 1, pp. 6–14, 2016.

[6] M. Sharifi, S. S. Kashi, and S. P. Ardakani, “LAP: a lightweight authentication protocol for smart dust wireless sensor networks,” in Proceedings of the International Symposium on Collaborative Technologies and Systems (CTS ’09), pp. 258–265, Baltimore, Md, USA, May 2009.

[8] J.-C. Wang, C.-H. Lin, E. Siahaan, B.-W. Chen, and H.-L. Chuang,“ Mixed sound event verification on wireless sensor network for home automation,”IEEE Trans. IndustrialInformatics, vol. 10, no. 1, pp. 803- 812, Feb. 2014.

[9] David Martins and Hervé Guyennet. Wireless Sensor Network Attacks and Security Mechanisms: A Short Survey, In Network-Based Information Systems (NBiS), 13th International Conference on (pp. 313- 320). IEEE, 2010.

[10] P.Samundiswary, D.Sathian and P. Dananjayan. (2010). Secured greedy perimeter stateless routing for wireless sensor networks, International Journal of Ad hoc, Sensor &Ubiquitous Computing(IJASUC)1, (2), 2010.

[11] Pathan, K., AI-S., Security of SelfOrganizing Networks-MANET, WSN, VANET,WMN. ISB N-13:978-1-4398-1920-3. Taylor and Francis Group, 2011.

[12] W. Shen, T. Zhang, F. Barac, and M. Gidlund, “PriorityMAC: A priority enhanced MAC protocol for critical traffic in industrial wireless sensor and actuator networks,”IEEE Trans.Industrial Informatics, vol. 10, no. 1, pp. 824-835, Feb. 2014.

[13] X. Zhang and J. Wang, “An efficient key management scheme in hierarchical wireless sensor networks,” in Proceedings of the International Conference on Computing, Communication and Security (ICCCS ’15), pp. 1–7, Pamplemousses, Mauritius, December 2015.

[14] Z. Wei, “Self-updating hash chains based on erasure coding,” in Proceedings of the International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE ’10), pp. 173–175, Changchun, China, August 2010.

[15] X.-Y. Yang, J.-J. Wang, J.-Y. Chen, and X.-Z. Pan, “A self-renewal hash chain scheme based on fair exchange idea(SRHC-FEI),” in Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT ’10), pp. 152–156, Chengdu, China, July 2010.

[16] Boukerche A. Werner Nelem Pazzia R., Borges Araujo R. "Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments"; Journal of Parallel and Distributed Computing, 4 : Vol. 66. - pp. 586-599, Avr 2006.

[17] Boukerche A., Chatzigiannakisb I., Nikoletseas S. "A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range, Vol. 29. pp477-489, 2006.

[18] D. J. Malan, M. Welsh, and M. D. Smith, "A Public-Key Infrastructure for Key Distribution in TinyOS based on Elliptic Curve Cryptography", Proc. 1st IEEE Int'l.Conf. Sensor and Ad Hoc Communications and Networks, Santa Clara, CA, Oct. 2004.

[19] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks", In Proceedings of the 9th ACM conference on Computer and communications security, November 2002.

[20] Daia F., Wub J. "On constructing k-connected k-dominating set in wireless ad hoc and sensor networks"; J. Parallel Distrib. Comput., Vol. 66. - pp. 947 – 958, 2006.

[21] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks", In IEEE Symposium on Security and Privacy, Berkeley, California, , pp. 197-213. May 11-14 2003.

[22] C. Qiu, H. Shen, S. Soltani, K. Sapra, H. Jiang, and J. O. Hallstrom, “CEDAR: a low-latency and distributed strategy for packet recovery in wireless networks,” IEEE/ACM Transactions on Networking (TON), vol. 23, no. 5, pp. 1514–1527, 2015.

[23] S. Zhu, S. Setia, and S. Jajodia, "Leap : efficient security mechanisms for large-scale distributed sensor networks", CCS 03 : Proceedings of the 10th ACM conference on Computer and communications security (New York, NY, USA), ACM Press, , pp. 62–72, 2003.

[24] Changlong Chen, Min Song, and George Hsieh. Intrusion detection of Sinkhole attack in large scale Wireless Sensor Networks, In Wireless Communications, Networkingand Information Security (WCNIS), 2010 IEEE International Conference on (pp.711-716). IEEE,2010.

[25] J. Deng, R. Han, and S. Mishra, "Insens: intrusion-Tolerant Routing for Wireless Sensor Networks", Computer Communications 29, no. 2, 216–230, 2006.

[26] M. H. Eldefrawy, M. K. Khan, and K. Alghathbar, "One-time password system with infinite nested Hash chains," in Security Technology, Disaster Recovery and Business Continuity, pp. 161–170, Springer, Berlin, Germany, 2010.

[27] J. Yin and S. Madria, "Secrout : A secure routing protocol for sensor networks", AINA 06 : Proceedings of the 20th International Conference on Advanced Information Networking and Applications (Washington, DC, USA), vol. 1, pp. 393–398, 2006.

[28] Chris Karlof, David Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures"; Ad Hoc Networks Vol.1, pp. 293–315, 2003.

[29] Josep Domingo-Ferrer. "A provably secure additive and multiplicative privacy homomorphism". In ISC '02 : Proceedings of the 5th International Conference on Information Security, 2002.

[30] Claude Castelluccia, Einar Mykletun, and Gene Tsudik. "Efficient aggregation of encrypted data in wireless sensor networks". In MobiQuitous, pages 109-117. 2005.

[31] L.Hu and D. Evans, "Secure aggregation for wireless networks",Workshop on Security and Assurance in Ad Hoc Networks, January 2003.

[32] Kemal Akkaya , Mohamed Younis "A survey on routing protocols for wireless sensor networks" Ad Hoc Networks 3, 2005.

[33] Mahimkar A et Rappaport, T.S, SecureDAV : A Secure Data Aggregation and Verication Protocol for Sensor Networks, 2004.

[34] M.Bagaa, N. Lasla, A. Ouadjaout and Y.Challal ; "SEDAN : Secure and Efficient protocol for Data Aggregation in wireless sensor Networks", 32nd IEEE Conference on Local Computer Networks pp. 1053-1060, Worksohop on Netwok Security, LCN 2007.

[35]M. Angjelichinoski, K. F. Trillingsgaard and P. Popovski, "A statistical learning approach to ultra-reliable low latency communica tion", *CoRR*, vol. abs/1809.05515, 2018.

[36] M. Emara, M. C. Filippou and I. Karls, Availability and reliability of wireless links in 5G systems: A space-time approach, Jun 2018.

[37] X. Zhou, M. R. McKay, B. Maham and A. Hjorungnes, "Rethinking the secrecy outage formulation: A secure transmission design perspective", IEEE Communications Letters, vol. 15, no. 3, pp. 302-304, March 2011.

[38] B. He, X. Zhou and A. L. Swindlehurst, "On secrecy metrics for physical layer security over quasi-static fading channels", IEEE Transactions on Wireless Communications, vol. 15, no. 10, pp. 6913-6924, Oct 2016.

[39] M. A. Gutierrez-Estevez, R. L. G. Cavalcante and S. Stanczak, "Non-parametric radio maps reconstruction via elastic net regularization with multi-kernels", 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pp. 1-5, 2018

[40] G. Oligeri, S. Chessa, R. Di Pietro, and G. Giunta, "Robust and efficient authentication of video stream broadcasting," ACM Transactions on Information and System Security, vol. 14, no. 1, article 5, pp. 1–25, 2011.

[41] L. Xu, M. Wen, and J. Li, "A bidirectional broadcasting authentication scheme for wireless sensor networks," in Proceedings of the IEEE Conference on Collaboration and Internet Computing (CIC '15), pp. 200–204, Hangzhou, China, October 2015.

[42] X. Li, N. Ruan, F. Wu, J. Li, and M. Li, "Efficient and enhanced broadcast authentication protocols based on multilevel μTESLA," in Proceedings of the 33rd IEEE International Performance Computing and Communications Conference (IPCCC '14), pp. 1–8, Austin, Tex, USA, December 2014.

[43] Y.-S. Chen, I.-L. Lin, C.-L. Lei, and Y.-H. Liao, "Broadcast authentication in sensor networks using compressed bloom filters," in Distributed Computing in Sensor Systems, pp. 9–111, Springer, Berlin, Germany, 2008.

[44] K. Ren, S. Yu, W. Lou, and Y. Zhang, "Multi-user broadcast authentication in wireless sensor networks," IEEE Transactions on Vehicular Technology, vol. 58, no. 8, pp. 4554–4564, 2009.

[45] C. Benzaid, S. Medjadba, A. Al-Nemrat, and N. Badache, "Accelerated verification of an ID-based signature scheme for broadcast authentication in wireless sensor networks," in Proceedings of the IEEE 15th International Conference on Computational Science and Engineering (CSE '12), pp. 633–639, Nicosia, Cyprus, December 2012.

[46] K. Ren, W. Lou, B. Zhu, and S. Jajodia, "Secure and efficient multicast in wireless sensor networks allowing adhoc group formation," IEEE Transactions on Vehicular Technology, vol. 58, no. 4, pp. 2018–2029, 2009

[47] X. Cao, W. Kou, L. Dang, and B. Zhao, "IMBAS: identity-based multi-user broadcast authentication in wireless sensor networks," Computer Communications, vol. 31, no. 4, pp. 659–667, 2008.

[48] R. D. Pietro, F. Martinelli, and N. V. Verde, "Broadcast authentication for resource constrained devices: a major pitfall and some solutions," in Proceedings of the 31st IEEE International Symposium on Reliable Distributed Systems (SRDS '12), pp. 213–218, Irvine, Calif, USA, October 2012.

[49] Gaurav Mehta *et al* , Security Techniques of WSN: A Review, International Journal of Computer Science and Mobile Computing, Vol.7 Issue.11, pg. 167-172, November-