Security Issues and Solutions related to Data Aggregation Process in WSN

Chérif Diallo,
Laboratoire Algèbre, Cryptographie, Codes et Applications (ACCA),
UFR Sciences appliquées et de Technologies (UFR SAT),
Université Gaston Berger, Saint-Louis, SENEGAL

Summary
In Wireless Sensor Networks (WSN), the nodes are often deployed to collect environmental data related to some particular relevant events. The collected data are then sent to a sink node or a base station following operations aimed at optimizing critical network resources such as storage, computing capacities and bandwidth. However, sensors have low processing and storage capabilities, small bandwidth and relatively low energetic resources. Thus, data aggregation is one of the most powerful techniques used to optimize sensor resources. Instead of sending data directly to the base station, the sensors send it first to an aggregation node that will compute a summary avoiding then redundant data and reducing the number of transmissions. During the data aggregation process, several attacks exploiting network vulnerabilities and weaknesses threaten sensor applications. In this paper we will give an overview study of the data aggregation process security issues and solutions in WSNs.

Key words:
Wireless Sensor Networks; Data Aggregation; Security; Cryptography.

1. Introduction

A sensor node is mainly composed by four basic units [Fig. 1]: the sensing unit, the processing unit, the wireless transceiver unit, and the power unit.

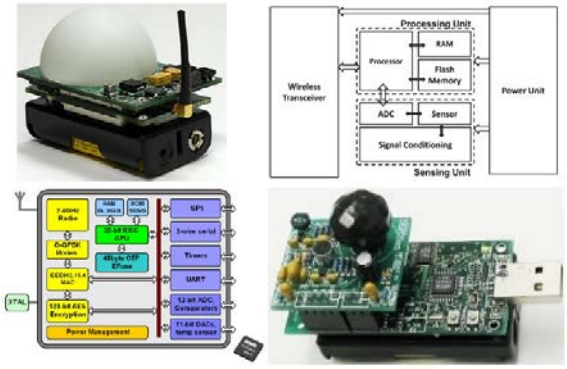

Fig. 1: Sensor node and its units

Thereby, a Wireless Sensor Network (WSN) [Fig. 2] consists of a set of several sensor nodes dedicated to the monitoring of a given environment (sensor field) for accurate application (temperature, light, pressure, control, monitoring, intrusion detection, air humidity, agriculture, home automation, medical field, etc.). Then sensors communicate with each other to relay information to a Sink node or a Base Station that communicates with the user interface.

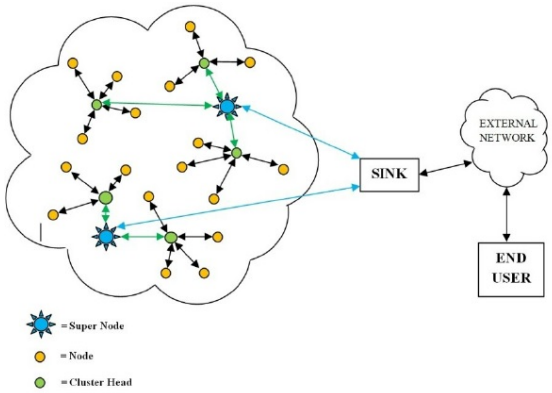

Fig. 2: Illustration of a Wireless Sensor Network (WSN)

Sensor applications are numerous and cover almost all areas. Nevertheless, whatever the domain concerned, the application must be designed to meet quality of service and security constraints.

Thus, beyond the security problems associated with the sensors themselves and the data exchanged between these nodes, it is essential to also guarantee the security of the operations required for the proper functioning of the network, such as neighborhood discovery process, routing calculation, clusters formation, and data aggregation operations. Energy constrained sensors have also low computational capability and low storage capacity. Moreover, the data collected by each node must be sent to the base station for processing in order to help a human operator in taking good decision. Thus, aggregation techniques [Fig. 3, 4] are used in some WSN applications

Manuscript received April 5, 2017
Manuscript revised April 20, 2017

to reduce communications in the network and then improve energy saving.

In [1], Becker et *al.* have defined the aggregation as follows: *"Aggregation Methods are general methods for which we try to decompose a complex system into subsystems that are simpler to study"*. Thus, [1] proposes a method of solving complex systems in two steps:

- ***Aggregation:*** solving each subsystem individually, and independently of other subsystems.
- ***Disaggregation:*** solving the global system taking into account the results obtained at the aggregation step.

Given their energy efficiency benefits, the two most important aggregation processes for sensor networks are data aggregation and clustering formation [2].
On the other hand, we know that sensors are characterized by their very limited lifetime due to the weakness of their batteries. Thus, the main aim in WSN is to reduce energy consumption by eliminating the redundancy of transmitted data. This reduces the number of transmissions by combining data arriving from the different nodes in few aggregation points. Therefore aggregation methods help in increasing the network lifetime. Depending on the type of sensor application, data aggregation is a fundamental mean to solve the problem of redundant data transmission. Indeed, in a cold chain monitoring application, several nodes may be required to send the same information related to a same coverage area. By applying data aggregation operations, the aggregation nodes eliminate redundant data so as to considerably reduce the overall amount of transmitted data while transmitting the useful information [2]. Thereby, the questions to be answered by the principle of data aggregation are:

- Which are the aggregation functions to use? How to categorize them?
- How to form the aggregation tree, where should the aggregation nodes be placed?
- How long time will it take for an aggregation node to collect information before applying the aggregation function?
- What should an interface look like to easily express the actions of aggregation process?

As a possible answer of the first above question, the aggregation function could be expressed, for example, by appropriate SQL requests [3]:

SELECT {agg(expr), attributes} ***FROM*** sensors
WHERE {selectionPredicates}
GROUP BY {attributes}
HAVING {havingPredicates}
EPOCH DURATION i

There are many categories of aggregation functions among which we could enumerate [2]:

- ***Duplicate sensitive:*** These functions are sensitive to duplication. Indeed, the result of the function is altered if the value measured by a node is taken into account more than once, in the calculation of the function. This category includes, for example, the SUM, MEDIAN and AVERAGE functions.
- ***Summary:*** A function is a summary type function, if its result strictly depends on the overall set of values recorded by the nodes. For instance, the ***SUM*** function.
- ***Exemplary: MIN*** and ***MAX*** are in the exemplary function category.
- ***Composable:*** An aggregation function f is in this category if the result of f applied to a set ω, can be known by applying f to a partition of ω via an auxiliary function g: $f(\omega) = g(f\ \omega_1), f(\omega_2))$ where $\omega = \omega_1 \cup \omega_2$ and $\omega_1 \cap \omega_2 = \emptyset$.

According to [3], the aggregation should be performed as close as possible to the source node in order to obtain more efficiency benefit. Moreover, an aggregation node should not wait too long before applying the aggregation functions. Because for a longer waiting time, the quantity of data that should be taken into account in the calculation of the aggregation function could be very large. This could result in additional energy losses. To avoid this problem, mechanisms must be designed to find a better compromise. The efficiency of data aggregation processing can be measured through different performance indicators:

- ***Accuracy:*** This is precision, strictness; it measures the difference between information received by the sink and the true information it should have received (since all data has not been transmitted).
- ***Completeness:*** This is the percentage of data included in the final aggregated data compared to the sensed data.
- ***Latency:*** Data aggregation can increase latency at the intermediate nodes that apply the data aggregation process.
- ***Reduced Message overhead:*** The advantage of data aggregation is that it reduces the amount of data transmitted over the network, saving energy and thus increasing network life.

In summary, data aggregation is required in the WSN to minimize redundant transmissions and thus save energy. To perform an aggregation operation, an intermediate node must have access to the data transmitted by other nodes in order to calculate the useful information by using an aggregation function [Fig. 3] such as: sum, average, maximum, minimum, etc.

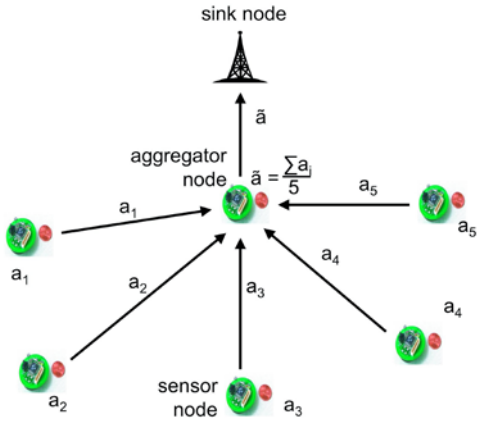

Fig. 3 : Data aggregation process in WSN

In a cluster-based data aggregation process [Fig. 4], the nodes are first grouped into clusters each managed by a cluster head. Thus, the aggregation function could be then computed either by one or more cluster heads or, failing that, by one or more regular nodes called aggregators.

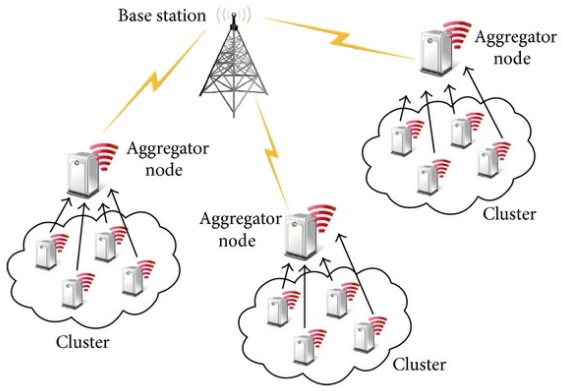

Fig. 4 : A cluster-based data aggregation process

In view of the importance of aggregation techniques for sensor networks, it is therefore essential to take into account the security issues and solutions in designing aggregation algorithms. However, many works addressing data aggregation do not address security issues and therefore WSN become vulnerable to some threats and attacks [4]. Without security protection, a malicious node could then insert false data into the network or falsify the result of an aggregation process. In this case, it could succeed in misleading the decision center which would consequently make poor decisions in relation to the values detected in an area by the sensors.

Apart from some specificity related to the nature of the sensors, WSNs share similar security requirements with traditional networks. The security needs of the data aggregation process in WSNs are thus the same as those in other ad hoc networks [5].

Data privacy: It ensures that the content of the information is not disclosed to an unauthorized entity. For [5], data privacy policy for the data aggregation process in WSN can be implemented in two ways: hop-by-hop, or end-to-end. In the first case, each aggregation node needs to decrypt the received data, apply an aggregate function, encrypt the aggregated data and then send it to a higher aggregation node towards the base station. This method has the main drawback to increase latency and data transmission delays. In the end-to-end model, the aggregation nodes apply the aggregation function on the received encrypted data using encryption techniques. For this method, latency, transmission delays and then energy consumption is reduced.

Data Integrity: it means that message content is not intentionally or accidentally altered during transmission. An adversary can compromise an aggregation node and succeed in inserting false data that will be sent to the base station. Moreover, after having compromised an aggregation node, the attacker could apply its own aggregation functions, or knowing the aggregation function, it could manipulate the parameters or inputs of this function. For example, assume that an aggregation function f which is computed by the compromised aggregation node as the mean of the entries x_i, $1 \leq i \leq n$: $f(x_1, \ldots, x_n) = (x_1 + \ldots + x_n) / n$. An attacker could falsify the result of the data aggregation process, by changing this function f by $f(x_1, \ldots, x_n) = \max \{ x_i, 1 \leq i \leq n \}$.

Data Freshness: This property guarantees that received data are recent, up-to-date and are not derived from old data that would be replayed. The freshness of the data will make it possible to protect the network against replay attacks. Without the implementation of data freshness techniques, an attacker could then intercept data and replay it in order to disrupt the aggregation process. This could result in manipulating aggregation results, cryptographic keys, etc.

Data Availability: It ensures that the network is properly functioning to satisfy all legitimate queries and that data are also accessible when needed by legitimate users and sensors. To ensure high availability for data aggregation operations, the security policy could contain the following mechanisms [5]:

Self-healing: It is a mechanism which consists in real-time scanning of the network in order to be able to detect the presence of a malicious node and then initiate corrective

operations and solutions intended to isolate that disruptive node.

Aggregation nodes rotation: performs regular changes of the aggregation nodes to more or less evenly distribute energy loads between the different nodes of the network.

Authentication: allows a recipient to verify that the received message truly originates from the legitimate source. Before taking data into account, the aggregation nodes must verify their authentication properties in order to avoid false aggregation results. Moreover, fighting against some attacks imply that each entity participating in the data aggregation process must have a cryptographic key enabling it to be authenticated by other nodes.

In this paper, we give an overview study of security issues and solutions of the data aggregation process in WSN. The rest of this paper is organized as follows. In the next section, we present a brief summary of the main WSN attacks and, in the third and last section, we will present data aggregation security systems.

2. Main attacks against data aggregation process

Given its weaknesses and its much vulnerability, sensor networks are subject to numerous attacks aimed at destroying efficiency, reliability, quality of service and security of applications. In [6], we give an overview study of the main attacks. The performance of the data aggregation process is closely related to the efficiency of the routing operations. Therefore, in this section we complete this survey with a short description of the main attacks targeting routing in sensor networks.

2.1 Replay attack

An attacker could collect data traffic that he will replay at a later time and succeed in affecting the aggregation results.

2.2 Stealthy attack

In this type of attack, the attacker could insert erroneous data into the network without revealing his own identity. This could give him an opportunity to successfully divert the aggregation results.

2.3 Selective forwarding attack

The selective forwarding attack could seriously affect the aggregation process because a compromised node may not send its data to the aggregation node or in the case of a compromised aggregation node, it could choose not to forward the aggregation results towards the base station.

2.4 Sybil attack

The Sybil attack in which an attacker has more than one identity could affect the data aggregation process in different ways [9]:

- Several identities could lead to the election of a malicious node as an aggregator.
- An attacker could generate multiple inputs with different sensed data in order to distort aggregation results.
- Some schemes use witnesses to validate aggregation and data will be considered valid only if ***n*** witnesses accept the aggregation result. In the presence of a Sybil attack, it would be possible to generate ***n*** or more witness identities to have the aggregation data accepted by the base station.

2.5 Node capture attack

A node capture by an attacker could give him the possibility to manipulate data, to extract cryptographic keys, to change the inputs of the aggregation functions or the functions themselves.

2.6 Jamming attack

A jamming attack could disrupt communications in such a way that an aggregation node may become unable to properly receive data to aggregate or prevent the base station from receiving the aggregated data.

2.7 Routing table overflow attack

In this kind of attack, a malicious node will advertise (to the authorized or legitimate sensors present in the network) routes concerning non-existent or illegitimate destinations. Because of low sensors processing and storage capabilities, the main objective of such an attack is to cause an overflow of the routing tables, which would in turn prevent the creation of legitimate entries corresponding to new routes towards authorized or legitimate nodes [7]. When this attack occurs, the base station may not properly receive the aggregated data.

2.8 Routing table poisoning attack

In this attack, one or more compromised nodes in the network could send fictitious routing updates or modify genuine route update packets sent to other non-compromised nodes. Routing table poisoning may result in sub-optimal routing, congestion in some sub-networks, or even make some parts of the network inaccessible [7].

With this kind of attack, the base station and aggregation nodes may not properly receive the aggregated or sensed data.

2.9 Routing cache poisoning attack

In the case of on-demand routing protocols (such as the AODV protocol), each node maintains a route cache which holds information regarding routes that have become known to the node in the recent past. Similar to routing table poisoning, an adversary can also poison the route cache to achieve similar objectives [7].

2.10 Rushing attack

On-demand routing protocols that use duplicate suppression during the route discovery process are vulnerable to this attack. An adversary node which receives a Route-request packet from the source node floods the packet quickly throughout the network before other nodes which also receive the same Route-request packet can react. Nodes that receive the legitimate Route-request packets assume those packets to be duplicates of the packets already received through the adversary node and hence discard those packets. Any route discovered by the source node would contain the adversary node as one of the intermediate nodes. Hence, the source node would not be able to find secure routes, that is, routes that do not include the adversary node. It is extremely difficult to detect such attacks in ad hoc wireless networks [7]. When this attack occurs, it make it possible to mount several other attacks such as passive listening, Man in the Middle, Sinkhole, Wormhole, Black hole, Sybil, Selective forwarding, and so on. Consequently, overall data aggregation process could be affected by this attack.

3. Data aggregation security schemes

There are many data aggregation security solutions which could be classified in different categories.

Paper [5] proposes a classification based on the number of aggregation nodes by classifying schemes according to two models: the one-aggregator model and the multi-aggregator model. In the one-aggregator model, the collected data are firstly sent to only one aggregator node which then computes aggregation operations before sending the summary towards the base station. In the multi-aggregator model, the collected data are aggregated more than once before reaching the base station.

Other possible classification type could take into account the cryptographic point of view. Thus, we can distinguish plaintext based techniques and those centered on the encrypted data.

3.1 Schemes based on the number of aggregation nodes

One can subdivide the nodes in a WSN in three subsets: the subset $\boldsymbol{C}$ set of sensors that collect event information, the subset $\boldsymbol{A}$ set composed of the aggregation nodes that receive the data collected by each sensors $\boldsymbol{S_i} \in \boldsymbol{C}$. Then, each $\boldsymbol{A_i} \in \boldsymbol{A}$ will apply the aggregation functions and then send the resulting data summary towards a base station $\boldsymbol{B_i} \in \boldsymbol{B_s}$. Where $\boldsymbol{B_s}$ is the subset of the base stations located in the network. As we have mentioned above, secure data aggregation protocols can be divided into two models based on the number of aggregation nodes [5]: the single-aggregator model and the multi-aggregator model. For each model, [5] evaluates the existence or absence of an integrity verification phase of the aggregated data.

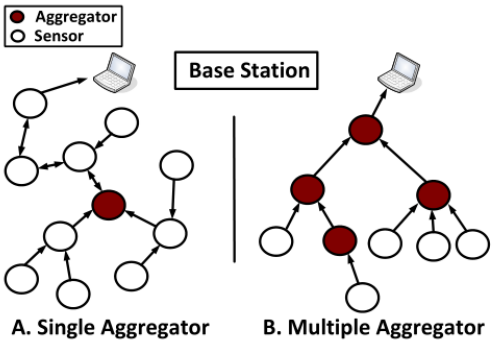

Fig. 5 : Models with one and multiple aggregators [5]

In this case, every data from each sensor is sent to the same aggregation node which will apply aggregation function before sending summaries towards a base station [Fig. 5-A]. To ensure data availability, the aggregation node must be a special one with high power capability in order to be able to support all expected packets and communications [9]. We can see that, the main goal of data aggregation protocols is not fully satisfied by this model because redundant data are always crossing the network, due to the presence of only one single aggregation node in the network. Thus, it is not suitable for large scale networks. However, in some cases, it remains applicable in small networks in which there is high data redundancy.

3.2 Schemes with multi-aggregator nodes

For this model, the sensed data are aggregated more than once before reaching the base station [Fig. 5-B]. Each aggregation node which receives data will first apply an aggregation function and then retransmits it either directly to the base station or to another aggregation node. This

model is more suitable for large scale networks with lot of redundant data.

3.3 Hop-by-hop data aggregation protocols

In techniques focused on unencrypted data, the aggregation function is performed by the aggregation node on unencrypted data [10]. In other words, the aggregation node must firstly decrypt the encrypted received data before secondly applying the aggregation function and finally encrypt the resulting summary in order to send it to a base station or to another aggregation node. These techniques are also called hop-by-hop encryption in data aggregation process. The main advantages of hop-by-hop encryption schemes are: (i) ensure network security at start up, (ii) perform in-network aggregation, and (iii) could give integrity protection of the data [11].

3.3.1 Key Distribution in Hop-by-hop data aggregation protocols

To achieve a hop-by-hop encryption in securing data aggregation process, it is possible to do keys distribution management in two ways according to the topology of the sensor network. In distributed WSNs or flat networks where all the nodes play the same role, the solution is to use a shared secret key between each pair of nodes. For hierarchical wireless sensor networks organized as clusters, the solution consists in distributing at each cluster a shared group key between the cluster head and the regular nodes of its cluster.

In the first case, different key distribution schemes are proposed. One solution is to pre-distribute a key to all the nodes before deploying the network [11], and after each deployment, each pair of nodes uses that key to generate another key which will be used to establish a communication. The other solution is to configure at each sensor $n - 1$ keys, each of which is shared with another sensor, for a network size of n nodes. There are also random key distribution schemes as described in [12].

For the second case, [12] proposes a solution which consists in sharing a cluster key at each cluster level. For this, each node u wishing to share a secret key with a neighbor v will first generate the key, encrypt it with the cluster key $K_{u,v}$ and then broadcast the encrypted key. The node v can then decrypt the received encrypted key, store it and then send its secret key to u. Another scheme uses elliptic curves for keys management. It consists in pre-configuring at each node an elliptic curve which will be used by that node to generate a public/private key pair. After network deployment, each node will then broadcast its public key.

3.3.2 Scheme of Hu et *al.* [13]

To ensure data integrity, many existing data aggregation security schemes use procedures which are specifics to these schemes but in most of the cases the key distribution phases remain similar to those described earlier. In [13], authors adopt in their aggregation scheme a method for ensuring the integrity of the data in which each node A is pre-configured with a key $K_{A,S}$ that it shares with the base station. They use a tree structure in which each node has a parent, the leaves collect the data and the internal nodes perform the aggregation process [Fig. 6].

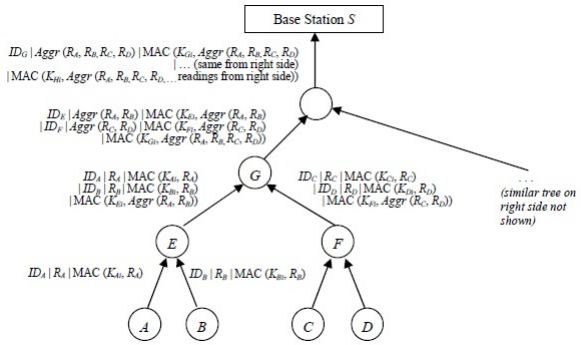

Fig 6 : Hu et al. Merkle tree [13]

In its i^{th} transmission step, the leaf A uses an encryption function E and the key K_{AS} to calculate a temporary key $K^i_{AS} = E(K_{AS}, i)$. It then sends to its immediate parent its identity ID_A, the collected data R_A and the encrypted message $MAC\ (K^i_{AS}, R_A)$. The parent B of each node A computes data aggregation of its child nodes and in turn sends the aggregation results $Aggr$, its identifier ID_B and the message $MAC\ (K^i_{BS}, Aggr)$ to its immediate parent. The process is repeated until all aggregation results arrive at the base station. Upon reception, the base station initiates the verification phase by broadcasting the keys according to the $\mu Tesla$ protocol and each aggregation node can subsequently verify the data which it has earlier aggregated.

As we can see, the scheme proposed by [13] consists of two phases: a data transmission phase and a verification phase. They adopt a delayed aggregation technique to prevent data disclosure which may result from key retrieval at a node by an attacker. In delayed aggregation, the data received by the nodes of the level k will be transmitted to the level $k-1$ nodes which will take care of the aggregation process. Once the aggregation results are received by the base station, it discloses the keys used for data encryption in the manner of $\mu Tesla$. After that, the nodes can check the aggregated data using the keys given by the base station. After verification, the nodes will inform the base station of the validity or non-validity of the aggregation

results. The base station can then validate or deny the aggregation results.

This scheme ensures data integrity but not data privacy because at a certain level, the aggregation summary is transmitted in unencrypted data. Moreover, a node capture attack against any node and its parent is sufficient to break data integrity. Otherwise, the nodes must store the data in a buffer memory until the keys are disclosed by the base station, which could be a real drawback due to the low storage capabilities of the nodes.

3.3.2 The SIA protocol

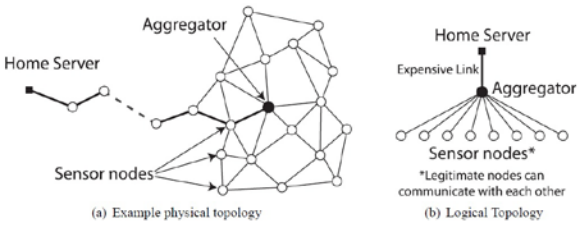

Fig. 7 : SIA protocol with ingle aggregator node [14].

The SIA protocol is a Secure Information Aggregation scheme in sensor networks proposed by [14]. It is designed to respond to queries about the sensed data. The validity of the data is verified by the base station according to a random sampling and an iterative proof. The scheme assumes the existence of a single aggregation node [Fig. 7] that authenticates the data by constructing a Merkle tree [Fig. 8]. Depending on the sampling mechanism, the scheme contains several aggregation algorithms using different aggregation functions such as median, mean, maximum and minimum.

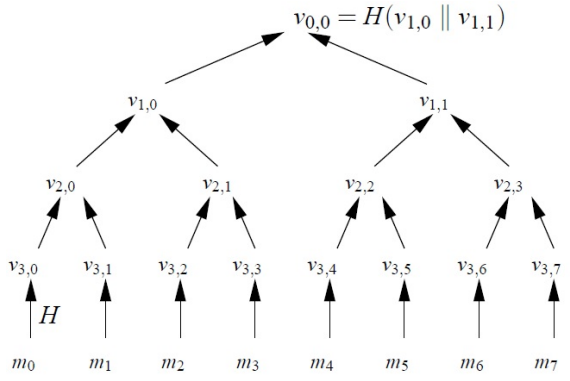

Fig. 8 : Merkle Hash Tree of SIA protocol [14].

In their proposal, all data collected by a node form the leaves in the Merkle tree and each node shares a key with its aggregation node. The aggregation node uses a ***H*** hash function and the leaves to build the nodes of the Merkle tree. The aggregation node starts by calculating the parent $v_{k,i}$ of each leaf m_i by $v_{k,i} = H(m_i)$. Then each internal node $v_{k-1,i}$ is computed from its left child $v_{k,i}$ and its right child $v_{k,i+1}$ by $v_{k-1,i} = H(v_{k,i} \mid v_{k,i+1})$. This process is repeated until $v_{0,0}$ is successfully computed. According to this method, each aggregation node could authenticate the aggregated data by verifying that the root $v_{0,0}$ of the tree can derive from the leaves $m_i , i \in [0, N]$.

The SIA protocol proposes the approach of forward secure authentication to ensure that even if an attacker corrupts a sensor node at a point in time, it will not be able to change any previous readings the sensor has recorded locally [14]. So it ensures data integrity and privacy, authentication and data freshness features. Moreover, it is particularly efficient against stealthy attacks.

3.3.3 The SDAP protocol

The SDAP protocol is a Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks proposed by [15]. This scheme has a mechanism to represent the network as a tree. The authors choose a probabilistic technique of dynamic partitioning of the tree into sub-trees, each of which holds a leading node. The figure [Fig. 9] shows an example of the SDAP aggregation tree. The nodes x, y and w" with the color dark gray are leader nodes, and the BS as the root is a default leader [15]. In this diagram, each leaf node sends its sensed data to its parent which computes a first data aggregation operation before sending the aggregation results to the leader node of its group. That leader node then applies a second data aggregation operation and finally sends the aggregation summary to the base station. The data encryption is performed using shared secret keys between each pair of nodes, which allows SDAP to ensure data integrity, confidentiality and also source node authentication.

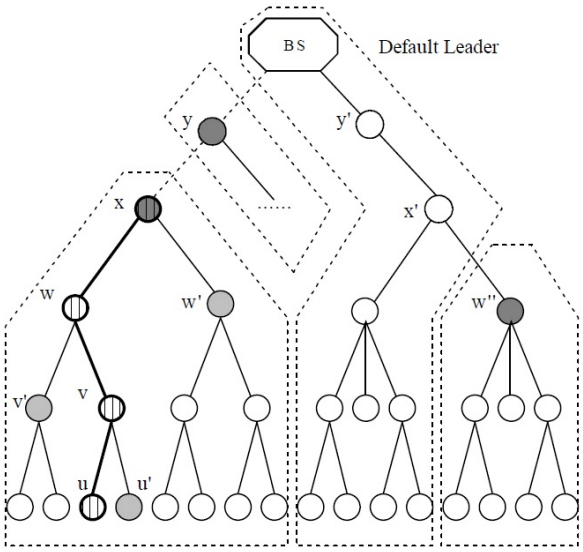

Fig. 9 : SDAP aggregation tree [15].

3.3.4 The Secure Hierarchical In-Network Aggregation scheme

The Secure Hierarchical In-Network Aggregation scheme proposed by [16] is inspired by the SIA model. Unlike SIA, which uses a single aggregation node, this scheme performs the data aggregation function according to several aggregation nodes. The aggregation process is initiated by the base station which broadcasts a query over the entire network, thus triggering the construction of an aggregation tree which looks like a hash tree [Fig. 10]. Each node sends its data to its father who performs the aggregation process until all aggregation results arrive at the base station. This latter performs a final data aggregation operation and then diffuses the result. This enables each node to check whether its sensed data have been truly added to the final result. Then each sensor confirms the final result by sending authentication codes which are aggregated according to an XOR function and then routed towards the base station which accepts the final aggregation result by verifying that all the sensors have sent an authentication code.

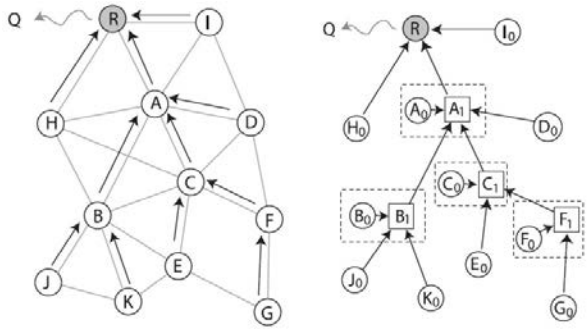

Fig. 10 : Secure Hierarchical In-Network Aggregation [16].

The Secure Hierarchical In-Network Aggregation algorithm is guaranteed to detect any manipulation of the aggregation result by an eventual adversary beyond what is achievable through direct injection of fictitious data at compromised nodes. In other words, the adversary can never gain any advantage from misrepresenting intermediate aggregation computations [16].

3.3.5 The ESPDA protocol

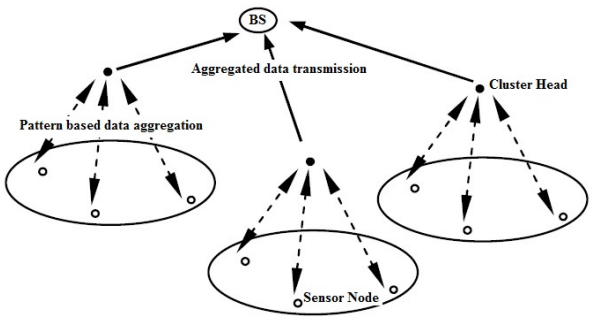

Fig. 12 : Data Transmission using ESPDA technique [17].

The ESPDA protocol is an Energy-efficient Secure Pattern Based Data Aggregation proposed by [17]. This secure aggregation scheme is suitable for hierarchical clustered wireless sensor networks. It is based on the pattern codes to aggregate the data. The pattern codes are representative data derived from the sensed data and make it possible to characterize the sensed data themselves [Fig. 12]. The nodes generate and send the pattern codes to the cluster heads for review control before allowing only one node to send to them the data representing the pattern codes. Finally, the cluster heads transmit these data towards the base station. The security mechanism adopted is described as follows: the base station is configured with the pairs (ID_i, K_i) which represent the identifier, the secret key of the node number i and a key K which it shares with all the nodes of the network. For each time interval, the base station broadcasts a K_b session key encrypted by $E_k(K)$ encryption function. The sensors then calculate their session secret key $K_{i,b} = K_b \Phi K_i$. The data is thus transmitted by the nodes by adding their ID_i to the MAC computed by $MAC\ (K_{i,b}, data)$.

ESPDA has a previous version which is a hybrid protocol at the borderline between hop-by-hop and end-to-end models, because in this previous version [18], cluster heads are not required to decrypt or encrypt the data received from the sensor nodes. The data aggregation is done before the actual data is transmitted by the sensor nodes. The sensor nodes have a unique secret built in key. The base station, periodically broadcasts a session key (different from pattern seed used in ESPDA) to maintain data freshness. The sensor node computes a node-specific-secret-key (NSSK) using the session key and the built in key [18]. This NSSK is used to encrypt and decrypt all the consequent data transmission during that session. The base station knows all the unique built in keys of the sensor nodes which are used to compute NSSK at the base station for decryption [18]. As we can see, the main drawback of this version is that too much responsibility is offered to the base station. Its compromise or its least failure could completely annihilate the security of the data aggregation process.

3.3.2 Discussions

The unencrypted data based techniques are a simple way to ensure the security of data aggregation in WSNs. In this type of techniques, data authentication is verified by all nodes participating in the aggregation process, which makes it possible to be protected against message insertion attacks and thus to ensure that the aggregated data are reliable.

Nevertheless, in these protocols, the need for the aggregation nodes to decrypt the received data before

applying the aggregation functions is a real security issue. Indeed, the node capture attack against an aggregation node which has just finished performing the decryption operation of the received data could give to the attacker an access to these data. The main difficulty thus remains the fact of wanting to ensure data privacy combined with an aggregation process in the network (that is to say *in-network aggregation*).

On the other hand, data encryption and decryption operations could reduce sensor resources, increase latency and bring about energy over-consumption. Moreover, these security mechanisms require nodes to store encryption and decryption keys, which could affect the limited memory storage of the sensors.

3.4 End-to-end data aggregation protocols

We have just seen that the unencrypted data based aggregation techniques have, all the same, some disadvantages that should be corrected. An alternative to this would be to use techniques centered on encrypted data which are called end-to-end data aggregation techniques. Instead of doing hop-by-hop encryption in data aggregation processing, end-to-end data aggregation or encrypted data aggregation techniques (i.e. *Concealed Data Aggregation: CDA*) are mechanisms in which aggregation functions are computed on encrypted data. Only the base station will decrypt the received aggregated data. To ensure this feature, end-to-end techniques use the concept of homomorphic encryption. Homomorphic encryption allows us to compute aggregation functions directly on encrypted data. For better understanding, let us give a formal definition of a homomorphic cryptosystem.

Definition: Homomorphic cryptosystem

Let Q and R be two sets; + and * the addition and multiplication operations respectively defined on these sets; and K the key space. Let us denote by $E: K * Q \rightarrow R$ the encryption operation; and by $D: K * R \rightarrow Q$ the decryption operation. For $a, b \in Q$ and $k \in K$, we say that the cryptosystem (E, D) is additively homomorphic if:

$$a + b = D_k(E_k(a) + E_k(b))$$,

We say that it is multiplicatively homomorphic if:

$$a * b = D_k(E_k(a) * E_k(b))$$

3.4.1 Key Distribution in end-to-end data aggregation protocols

Key distribution methods aggregation in end-to-end data aggregation techniques are similar to those found in unencrypted data based aggregation. Although the decryption processing is performed at the base station, it is nevertheless necessary in certain cases that the nodes share secret keys. For example, in a hierarchical network where the nodes send their data to a cluster head, it is necessary to set up a mechanism to secure the communications between the nodes and their cluster head. In [19], at the level of each cluster, the nodes and the cluster head share a cluster key used by the nodes to generate signatures of their data that they will send to their cluster head. The latter uses the same key to verify the signature of the aggregated data. The authors used elliptic curves to manage this key establishment phase in the network. On the other hand, it is also possible to use the methods proposed in [12],[20] to manage key distribution.

3.4.2 Data Integrity feature in end-to-end data aggregation protocols

Multiple works on secure end-to-end data aggregation techniques address the privacy issues related to unencrypted data based aggregation solutions. Indeed, in end-to-end data aggregation techniques, although the aggregation function is performed in the network, only the base station is authorized to decrypt the data and therefore no unencrypted data is transmitted over the network. If the problem of confidentiality is addressed, however, there remains the question of integrity which requires effective solutions.

To solve data integrity problems, different schemes are proposed, some of which sometimes depend on the network structure. For data encryption, many protocols use the encryption function proposed by [21]. The Domingo-Ferrer encryption function is a probabilistic encryption scheme in which the encrypted output of a message is randomly chosen from multiple encrypted data. The public input parameters of this function are an integer $d \geq 2$ and an integer g which is very large and which has several small divisors. There must be also several integers less than g and invertible modulo g. In modular arithmetical, the modular inverse of a relative integer a for multiplication modulo n is an integer u satisfying the equation $a = 1 \bmod n$. Upon these conditions, the private key is given by:

$$k = (r, g')$$

The secret parameter $r \in Z_g$ is chosen such that $r^{-1} \bmod g$ exists when $\log_{g'} g$ indicates the security level of the encryption function. Then the set of unencrypted messages is Z_g' and the set of encrypted is Z_g^d. The encryption and decryption operations of a message a are given by:

Encryption: Divide randomly a by $a_1, \ldots, a_d$ such that:

$$a = a_1 \bmod g' + \ldots + a_d \bmod g'$$

and calculate:

$$E_k(a) = (a_1 r \bmod g\ ,\ a_2 r^2 \bmod g\ ,\ \ldots\ ,\ a_d r^d \bmod g)$$

Decryption: To find a_j, and to retrieve the value of a, calculate the j^{th} term by $r^{-j} \bmod$

$$D_k(E_k(a)) = a_1 \bmod g' + \ldots + a_d \bmod g'$$

In [22] and [23], this same data encryption function is used to ensure integrity. In these diagrams, it is assumed that the nodes and the base station share a private secret key K which is unknown by the aggregation nodes [11]. The sensors S_i, $1 \leq i \leq n$ collect the data s_i and send the encrypted data $s'_i = E_K(s_i)$ to the aggregation nodes. The latter apply their aggregation function to the encrypted data $f(s'_1, \ldots, s'_n)$ and send the result to the base station which carries out the verification according to this same encryption function.

In [19], data integrity is ensured using a Merkle scheme. The sensors transmit their data with their hashes to their cluster head which constructs a Merkle tree of the received hashes. The base station will check the signature of the data with its public key.

3.4.3 The CDA scheme

The CDA scheme is a Concealed Data Aggregation for Reverse Multicast Traffic proposed by [23]. This algorithm is a secure end-to-end data aggregation technique based on homomorphic encryption. The CDA scheme uses the Domingo-Ferrer encryption feature described above. In their protocol, the network is composed of sensor nodes ($S_1, \ldots, S_n$) that transmit sensed data, the aggregation nodes ($A_1, \ldots, A_l$) which compute data aggregation operations, and a base station (R) which decrypts the aggregation results. The authors consider that the parameters (r, g') are known by the sensors S_i and also by the base station. Each sensor S_i encrypts its sensed data s_i and then sends the encryption result $s'_i = E_{(r, g)}(s_i)$ to an aggregation node A_j. Then A_j applies its aggregation function f to compute $y' = f(s'_1, \ldots, s'_n)$ and passes the result y' to the base station R. The base station in turn decrypts the data by computing $y = D_{(r, g')}(y')$ to extract the result of the aggregation. The [Fig. 12] gives an illustration of this process. The authors show how applying CDA helps distributing the overall energy consumption in a balanced way and reducing the total energy loads in the network. Then, this reduces the risk of a disconnected WSN due to nodes with empty batteries.

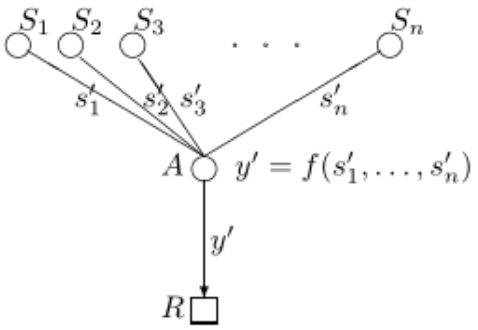

Fig. 13 : CDA: Secure end-to-end aggregation based on homomorphic encryption [23].

3.4.4 The SecureDAV scheme

Paper [19] proposes a Secure Data Aggregation and Verification protocol (SecureDAV) which allows the base station to only accept the correct sensed data. A key distribution mechanism is implemented immediately after the network deployment, allowing the nodes to share the necessary keys. The scheme uses a hierarchical structure in which each cluster head aggregates the data, calculates its mean, and then diffuses the aggregation result to all nodes in its own cluster. Each node compares the value received from the base station with its sensed data, and if the difference is less than a certain threshold, then the node generates a partial hash signature of the average which it sends to the base station. Then, the cluster head combines all the received signatures by forming a global signature which it finally transmits with the computed average to the base station. This one then checks the validity of the signature with its public key.

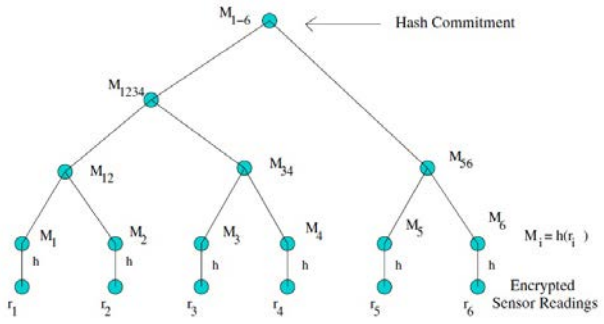

Fig. 14 : SecureDAV Merkle Hash Tree [19]

With SecureDAV, integrity check of the readings is done, and it ensures that the base station accepts the aggregate readings with high reliability, even if the cluster-head is compromised.

3.4.5 The Efficient Aggregation of Encrypted Data scheme

In [22], authors propose an efficient secure aggregation scheme, which also provides security and optimizes the network bandwidth. This scheme is based on a homomorphic stream encryption by replacing the ***XOR*** operations by the modular addition on the integers. With this encryption scheme, functions such as mean and variance can be used as aggregation functions. Each sensor encrypts its data x_i to obtain the encrypted $c_{xi} = Enc(x_i,k_i,M) = x_i + k_i \pmod M$ such that k_i is a randomly generated key ($k_i \in [0,M-1]$, M a very large integer). Upon reception of the encrypted data from all its child nodes, each aggregation node uses its aggregation function g to compute $z = g(c_{xi}, \ldots, c_{xn})$ and then send the result to the base station. This one decrypts the received encrypted data by calculating $Dec(z, K, M) = z - K \pmod M$, for $K = k_1 + \ldots + k_n$.

In [22] authors offer efficient and provably secure techniques for end-to-end privacy and authenticity. However, the scheme only supports mean and variance computation, but the same construction could be used as a building block for other aggregation protocol that support more advanced functions, such as median, mode, and range, etc.

3.4.6 The CDAP protocol

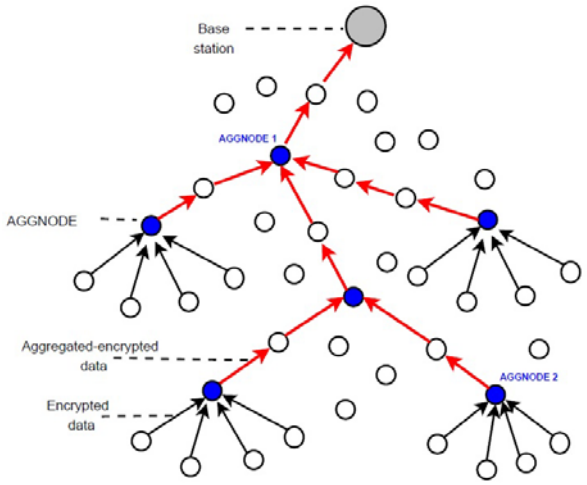

Fig. 15: The aggregation scenario of CDAP protocol. AGGNODEs collect information from their neighborhood and encrypted data are aggregated at AGGNODEs while data travels towards the base station [24].

The CDAP protocol is a Concealed Data Aggregation scheme using Privacy Homomorphism proposed by [24]. It is also based on homomorphic encryption to ensure the security of aggregated data. The author states that symmetric homomorphic encryption used in some protocols such as [23] contains security problems because of the unique shared key between nodes. For this, it uses an asymmetric homomorphic encryption and because of the extra calculation costs, the scheme uses special powerful nodes called AGGNODE which have sufficient resources to perform the aggregation. After the deployment of the network, each AGGNODE establishes pairs of keys with the nodes around its neighborhood which can then send their sensed data in a secure way to the AGNODE following a symmetric encryption algorithm. When an AGGNODE receives its aggregation data, it decrypts it, aggregates it and encrypts the result before transmitting it to the base station. This last one can then decrypt the aggregation result with its private key. In the CDAP protocol, the computational overhead imposed by the privacy homomorphic encryption functions is tolerated by employing a set of powerful nodes (AGGNODEs). So, the main drawback of this protocol is that, it is particularly intended for heterogeneous sensor networks.

3.4.7 The Secure Data Aggregation with Multiple Encryption scheme

In [25], Onen et *al.* propose a secured layer aggregation scheme based on additive homomorphic encryption. A pseudo-random key distribution mechanism is put in place allowing the nodes to share symmetric keys. Authors combine homomorphic encryption with a multiple encryption process. In this scheme, the network is structured as a tree in which each leaf sends its sensed data to its parent which extracts the shared key (with its child node), adds its sensed data and the secret shared key with its parent in the tree before sending the result to it. This process is repeated until all data arrives at the base station that performs the authentication of the received results. This aggregation scheme provides both generic and end-to-end confidentiality and is robust against bogus message injections and message losses [25].

3.4.8 Discussions

The Table 1 and the figure [Fig. 16] give a summary of the different data aggregation protocols presented in this paper.

As we can see, aggregation techniques centered on encrypted data has proven to be an efficient and reliable way of securing data aggregation in wireless sensor networks. With these techniques, the data can be aggregated in the network while guaranteeing their integrity and their confidentiality whereas hop-by-hop aggregation protocols do not fully support the privacy feature. The end-to-end data aggregation mechanism is made possible by exploiting the properties of a homomorphic cryptosystem that allows delegating the data aggregation processing to an operator which should not

have an access to the data. As a result, aggregation functions such as mean, sum, variance, and so on, are used by the protocols to aggregate the encrypted data that can only be decrypted by the base station. By doing so, even in the presence of a node capture attack involving an aggregation node, the attacker might not be able to access the data because only encrypted data circulates in the network. The other advantages rely on the sensor resources optimization because the nodes are no longer responsible for performing certain data encryption and decryption operations. Moreover, they no longer need to store certain encryption keys.

The main drawback of end-to-end data aggregation protocols remains the fact that they give a lot of responsibility to the base station. Thereby, in the presence of a node capture attack involving the base station, the attacker might be able to access the aggregated data [Fig. 13]. Moreover, these techniques do not fully support overleaps when the network is composed of many nodes. Finally, in the case of node authentication mechanisms are not implemented in the network, these protocols would not be resistant to Sybil or Wormhole attacks.

Conclusions

Data aggregation in WSN has become an effective technique for preserving the resources of the nodes. An adversary may attempt to break the aggregation process by mounting attacks that could affect the final aggregation result and then lead to misinterpretations. Nevertheless, many proposed aggregation protocols do not address security issues. In this survey, we have presented few different techniques used in securing the data aggregation process. There are mainly two families of techniques: the hop-by-hop data aggregation protocols and the end-to-end ones. In the first case, the data privacy is not fully supported and the data integrity can be ensured with a message authentication code, while in the second case, confidentiality and data integrity are ensured using for the latter a homomorphic encryption function. Moreover, the end-to-end data aggregation protocols consume fewer resources and are more resistant to most of the attacks, even if they remain vulnerable to Sybil or Wormhole attacks when node authentication mechanisms are not implemented in the network. Finally, it is deplorable that none of the works studied in this overview gives an estimate of energy over-consumption [26] induced by securing data aggregation operations.

Table 1 : Secure aggregation schemes

Scheme	Confidentiality	Integrity	Authentication	Freshness	Availability
Hu et *al.*		✗	✗	✗	
SIA	✗	✗	✗	✗	
SDAP	✗	✗	✗		
Chan et *al.*	✗	✗	✗		
ESPDA	✗	✗	✗	✗	
CDA	✗				
SecureDAV	✗	✗	✗		
Castelluccia et *al.*	✗		✗		
CDAP	✗				
Onen et *al.*	✗				✗

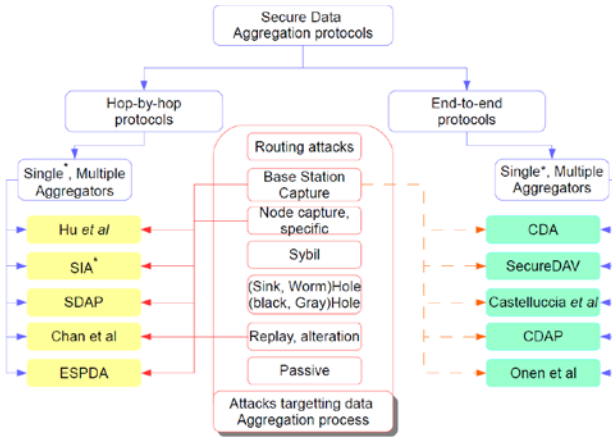

Fig. 16 : Secure data aggregation and attacks

Acknowledgments

The author would like to thank "Centre d'Excellence en Mathématiques, Informatique et TIC (CEA-MITIC)"; UFR Sciences appliquées et de Technologies (UFR SAT), Université Gaston Berger, Saint-Louis, Sénégal. (http://www.ceamitic.sn/).

References

[1] Becker, M. and Beylot, A.L. (2006) 'Simulation des réseaux', *In Traité IC2, Série Réseaux et Télécoms, Hermes.*

[2] Diallo, C. (2010) 'Techniques d'amélioration du routage et de la formation des clusters multisauts dans les réseaux de capteurs sans fil', *In PhD dissertation, Télécom & Management SudParis, Evry, France*.

[3] Karl, H. and Willig, A. (2005) 'Protocols and architectures for wireless sensor networks', *In Wiley*.

[4] D. Wagner. (2004) 'Resilient aggregation in sensor networks', *In Proceedings of the 2Nd ACM Workshop on Security of Ad Hoc and Sensor Networks, SASN'04, New York, NY, USA, pp.78-87*.

[5] Alzaid, H M. (2011) 'Secure data aggregation in wireless sensor networks', *In PhD dissertation, Queensland University of Technology*.

[6] Diallo, C., Sawaré, A. and Sow, M. T. (2017) 'Security Issues and Solutions in Wireless Sensor Networks', *In International Journal of Computer Science and Information Security, (IJCSIS) Vol. 15, No. 3, March 2017*.

[7] Manoj, B.S. and Siva Ram Murthy, C. (2004) 'Transport Layer and Security Protocols for Ad Hoc Wireless Networks', *In Ad Hoc Wireless Networks: Architectures and Protocols, Book ISBN-13: 978-0-13-147023-1, Published by Prentice Hall*.

[8] Maksud, S. S. and Patel, A. D. (2015) 'Secure data aggregation using homomorpic encryption in wireless sensor networks : A survey', *Advances in Computer Science and Information Technology (ACSIT), pp.13-20*.

[9] Alzaid, H., Foo, E. and Nieto, J. G. (2008) 'Secure data aggregation in wireless sensor network : A survey', *In Proceedings of the Sixth Australasian Conference on Information Security, Darlinghurst, Australia, Australian Computer Society, Inc., AISC '08, Vol. 81, pp.93-105*.

[10] Labroui, N. (2012) 'La sécurité dans les réseaux sans Fil ad hoc', *PhD dissertation, Université de Tlemcen.*

[11] Sang, Y., Shen, H., Inoguchi, Y., Tan, Y. and Xiong, N. (2006) 'Secure data aggregation in wireless sensor networks : A survey', ' *In Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2006), Taipei, Taiwan, pp. 315-320*.

[12] Zhu, S., Setia, S. and Jajodia, S. (2003) 'Leap: Efficient security mechanisms for large-scale distributed sensor networks', *In Proceedings of the 10th ACM Conference on Computer and Communications Security, CCS '03, New York, NY, USA, pp.62-72*.

[13] Hu, L. and Evans, D. (2003) 'Secure aggregation for wireless networks', *Workshop on Security and Assurance in Ad hoc Networks*.

[14] Chan, H., Perrig, A, Przydatek, B. and Song, D. (2007) 'Sia : Secure information aggregation in sensor networks', *J. Comput. Secur., Vol. 15(1), January 2007, pp.69-102*.

[15] Yang, Y., Wang, X., Zhu, S. and Cao G. (2006) 'Sdap: a secure hop-by-hop data aggregation protocol for sensor networks', *In MobiHoc, 2006, pp.356-367*.

[16] Chan, H., Perrig, A. and Song, D. (2006) 'Secure hierarchical in-network aggregation in sensor networks', *In Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS '06, New York, NY, USA, pp. 278-287*.

[17] Cam, H., Ozdemir, S., Nair, P., Muthuavinashiappan, D. and Sanli, H. O. (2006) 'Energy-efficient secure pattern based data aggregation for wireless sensor networks', *In Comput. Commun., February 2006, Vol. 29(4), pp.446-455*.

[18] Cam, H., Ozdemir, S., Nair, P., and Muthuavinashiappan, D. (2003) 'ESPDA: Energy-efficient secure pattern based data aggregation for wireless sensor networks', *In 2nd IEEE Conference on Sensors, Toronto, Canada, 2003*.

[19] Mahimkar, A., and Rappaport, T. S. (2004) 'Securedav : A secure data aggregation and verification protocol for sensor networks', *In Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'04), pp. 2175-2179*.

[20] Eschenauer, L. and Gligor, V. D. (2002) 'A key-management scheme for distributed sensor networks', *In Proceedings of the 9th ACM Conference on Computer and Communications Security, pp.41-47*.

[21] Domingo-Ferrer, J. (2002) 'A provably secure additive and multiplicative privacy homomorphism', *In Proceedings of the 5th International Conference on Information Security, ISC '02, London, UK, Springer-Verlag, pp. 471-483*.

[22] Castelluccia, C., Chan, A. C-F., Mykletun, E. and Tsudik, G. (2009) 'Efficient and provably secure aggregation of encrypted data in wireless sensor networks', *In ACM Trans. Sen. Netw., June 2009, Vol. 5(3)*.

[23] Girao, J., Westhoff, D. and Schneider, M. (2009) 'Cda : Concealed data aggregation for reverse multicast traffic in wireless sensor networks', *In Proceedings of IEEE International Conference on Communications, ICC2005, Seoul, Korea*.

[24] Ozdemir, S. (2007) 'Concealed data aggregation in heterogeneous sensor networks using privacy homomorphism', *IEEE, 8 2007*.

[25] Onen, M. and Molva, R. (2007) 'Secure data aggregation with multiple encryption', *In Koen Langendoen and Thiemo Voigt, editors, EWSN, volume 4373 of Lecture Notes in Computer Science, Springer, pp 117-132*.

[26] Sow, M. T. and Diallo, C., 'Energy Over-Consumption Induced by Securing Network Operations', *In Proceedings of The Second International Conference on Frontiers of Sensors Technologies, (IEEE-ICFST 2017)*, ISBN: 978-1-5090-4858-8/17/ ©2017 IEEE, pp: 154-160.

Dr. Chérif Diallo received his PhD in Computer Science at Telecom SudParis (France). CISSP certified, He is a Lecturer at the Department of the Applied Sciences and Technologies, University Gaston Berger of Saint-Louis, Sénégal. His research interests include issues related to wireless sensor networks, complex networks, Internet of Things including also the security of systems and networks. He is author of research studies published at national and international journals, conference proceedings as well as book chapters.