10

Secure Data Aggregation in Wireless Sensor Networks

Hani Alzaid
Queensland University of Technology
Australia
King Abdulaziz City for Science and Technology
Saudi Arabia

Ernest Foo and Juan Gonzalez Neito
Queensland University of Technology
Australia

DongGook Park
Sunchon University
Korea

Abstract
Recent advances in wireless sensor networks (WSNs) have led to several new promising applications including habitat monitoring and target tracking. However, data communication between nodes consumes a large portion of the entire energy consumption of the WSNs. Consequently, data aggregation techniques can significantly help to reduce the energy consumption by eliminating redundant data travelling back to the base station. The security issues such as data integrity, confidentiality, and freshness in data aggregation become crucial when the WSN is deployed in a remote or hostile environment where sensors are prone to node failures and compromises. There is currently research potential in securing data aggregation in WSNs. With this in mind, the security issues in data aggregation for the WSN will be discussed in this paper. Then, the adversarial model that can exist in any aggregation protocol will be explained. After that, the "state-of-the-art" in secure data aggregation schemes will be surveyed and then classified into two categories based on the number of aggregator nodes and the existence of the verification phase. Finally, a conceptual framework will be proposed to provide new designs with the minimum security requirements against a certain type of adversary. This framework gives a better understanding of those schemes and facilitates the evaluation process.

Keywords: Secure aggregation, wireless sensor networks, performance analysis, security analysis, survey.

1. Introduction

A WSN is a highly distributed network of small wireless nodes deployed in large numbers to monitor the environment or other systems by the measurement of physical parameters such as temperature, pressure, or relative humidity (Murthy & Manoj, 2004, p 647). Sensor nodes collaborate to form an ad hoc network capable of reporting network activities to a data collection sink. Recently, WSNs have been used in many promising applications including habitat monitoring (Mainwaring et al., 2002) and target tracking (He et al., 2006). However, WSNs are resource constrained with limited energy lifetime, slow computation, small memory, and limited communication capabilities (Yick et al., 2008). The current version of sensors such as mica2 uses a 16 bit, 8MHz Texas Instruments MSP430 micro-controller with only 10 KB RAM, 48 KB program space, 1024 KB external flash, and is powered by two AA batteries (Crossbow Technology Inc., 2006). Therefore, the energy impact of adding security features should be considered. For example, data authentication in TinyOS increases the consumed energy by almost 3% while data authentication and encryption by 14% (Guimarães et al., 2005). Furthermore, the processing capabilities in sensor nodes are generally not as powerful as those in the nodes of a wired network. Complex cryptographic algorithms are consequently impractical for WSNs.

Not only do the resource limitations affect the WSN performance, but also the deployment nature. Most WSNs are deployed in remote or hostile environments where nodes are exposed to physical attacks since anyone can access the deployment area. Moreover, since the WSNs are deployed in a remote environment, the only way to manage and control the network is via wireless communication, which makes any physical operation such as battery replacement difficult. Another factor that affects the performance of WSNs is communication instability due to the nature of the unreliable wireless communication. For example, if two sensors that have the same aggregator node start sending packets at the same time, conflicts will occur near the aggregator node and the transfer process will fail. In addition, packets might be dropped at highly congested nodes, since the packet based routing of the WSN is connectionless, which is inherently unreliable. As a result, any proposed protocol might also lose critical security packets such as keys, if it does not maintain a reasonable channel error rate. Finally, network congestion, multi-hop routing, node processing, and data aggregation introduce delays in the network and might lead to greater latency. Achieving synchronization between sensor nodes will, therefore, be difficult once latency is getting bigger. The synchronization issue can also be critical for data aggregation security since a part of the security scheme, such as key distribution, cannot work efficiently without achieving a low latency rate.

Due to these limitations, devising security protocols for WSNs is complicated and may not be successfully accomplished by the simple adaptation of security solutions designed for wired networks. Studies by Wagner (2004) and Krishnamachari et al. (2002) showed that data transmission consumes much more energy than computation. Data transmission accounts for 70% of the energy cost of computation and communication for the SNEP protocol (Perrig et al., 2002). Data aggregation can significantly help to reduce this consumption by eliminating redundant data. However, the aggregators are vulnerable to attack, especially if they are not equipped with tamper-resistant hardware. When an aggregator node is compromised, it is easy for the adversary to change the aggregation result and inject false data into WSNs. Unfortunately, the security mechanisms used in other

network environments are not appropriate for WSN domains, since they are typically based on public key cryptography which is too expensive for sensor nodes.
Secure data aggregation schemes are classified, in this chapter, based on how many times the data is aggregated during its travel to the base station. Our contributions in this chapter include the following:

- The secure data aggregation is defined informally and then the security issues in data aggregation for WSNs are discussed.
- An adversarial model, which can be expected in any secure data aggregation scheme, is proposed. This model covers different types of adversaries where the computational strength, the network access level, and node's secret-access level may vary.
- A survey of the "state-of-the-art" in secure data aggregation schemes is presented and these schemes are then classified into two groups according to the number of aggregator nodes, and whether the verification phase of the aggregated result is considered or not.
- Finally, the security and performance analysis of current secure data aggregation protocols are given and then a conceptual framework is proposed in order to establish common ground (or test-bed) to compare different secure data aggregation schemes. This framework also helps to draw the road map for the future design of attack resistant secure data aggregation.

The rest of the chapter is organized as follows: Section 2 gives introductory information about secure data aggregation in WSNs and discusses the security requirements for secure data aggregation protocols. Section 3 discusses different types of the expected adversarial model that threaten secure data aggregation protocols in WSNs. Section 4 surveys, in detail, some of the current secure data aggregation protocols and classifies them into two models. A security analysis of these protocols is discussed in Section 5. Section 6 discusses the performance analysis of these protocols. Finally, the chapter is concluded.

2. Secure Data Aggregation in Wireless Sensor Networks

In many applications, the physical phenomenon is sensed by sensor nodes and then reported to the base station. To reduce the energy consumption of the sensor nodes, these applications may employ in-network aggregation before the data reaches the base station. Compromised nodes can thus perform malicious activities which affect the aggregation results. Before these malicious activities are discussed, the motivation behind secure data aggregation in WSNs is explained, followed by the security requirements of WSNs required to strengthen attack-resistant data aggregation protocols.

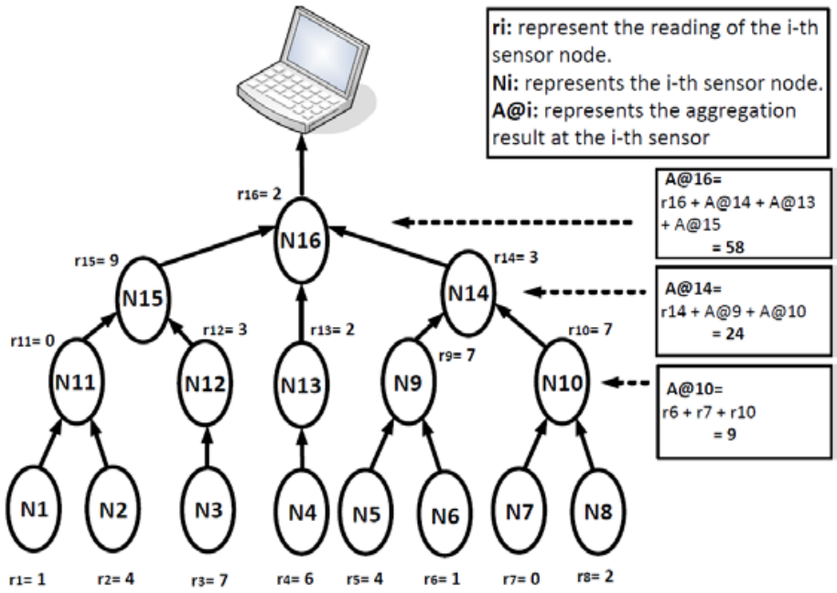


Fig. 1. An aggregation scenario using the SUM aggregation function.

2.1 Data Aggregation in Wireless Sensor Networks

Typically, there are three types of nodes in WSNs that perform in-network processing activities: normal sensor nodes, aggregators, and a querier. The aggregators collect data from a subset of the network, aggregate the data using a suitable aggregation function, and then transmit the aggregated result to an upper aggregator or to the querier who generated the query. The querier is entrusted with the task of processing the received sensor data and derives meaningful information reflecting the events in the target field. It can be the base station or sometimes an external user who has permission to interact with the network depending on the network architecture. Data communication between sensors, aggregators and the querier consumes a large portion of the total energy consumption of the WSN. For example, the WSN in Figure 1 contains 16 sensor nodes and performs SUM as the aggregation function in order to minimize the number of packets that are reported back to the base station, thus reducing the energy consumption. Node 1, node 2, ..., and node 8 are normal nodes that collect data and report them back to the upper nodes, whereas node 9, node 10, ..., and node 16 are aggregators that perform both sensing and aggregating activities.

In our example in Figure 1, every node will respond to a query and report its sensed information individually, and the total number of packets, reported back to the base station, would therefore be 50 packets if there was no in-network processing (or aggregation) capability. However, the number of packets drops to 16 if the in-network processing (aggregation) capability is enabled.

Most existing proposals for data aggregation are subject to attack (Wagner, 2004). Once a single node is compromised, it is easy for an adversary to inject false data into the network and mislead the aggregator to accept false readings. Because of this, the need for secure data aggregation is raised and its importance needs to be highlighted. However, the design principles for secure data aggregation schemes are poorly understood. There is no clear definition of what secure data aggregation should mean, what requirements they should

have, and what type of adversary they have to defend. Existing protocols might have one or more of the security requirements discussed in section 2.2 depending on what the secure aggregation looks like to the authors. Unfortunately, following this method to address the security in data aggregation is impractical. For example, Przydatek et al. addressed secure data aggregation in their protocol from the point of view of detecting forged data aggregation values (2003). This does not cover security issues such as how to elect aggregators or how to set up trust between aggregators and sensor nodes. Some protocols provide more security requirements than others, or send more bits than others as seen in Sections 5 and 6. There is no common ground that allows for comparison between different aggregation protocols.

Przydatek et al. defined secure data aggregation as "*the efficient delivery of the summary of sensor readings that are reported to an off-site user in such a way that ensures these reported readings have not been altered*" (2003). They considered an aggregation application where the querier is located outside the WSN and the base station acts as an aggregator. A detailed definition of secure data aggregation is needed for the sake of better understanding. Shi and Perrig highlighted the error sources that affect the aggregated data, and defined secure data aggregation as "*the process of obtaining a relative estimate of the sensor readings with the ability to detect and reject reported data that is significantly distorted by corrupted nodes or injected by malicious nodes*" (2004). However, rejecting reported data injected by malicious nodes consumes the network resources, specifically the nodes' batteries, since the malicious packet will be processed each time at the aggregator point. The damage caused by malicious nodes or compromised nodes should be reduced by adding a self-healing property to the network. This property helps the network in learning how to handle new threats through extensive monitoring of network activities, machine learning, and modelling of the network behaviour. Therefore, we take a step further and stipulate the main components of a robust secure data aggregation protocol as follows:

- Ability to provide fair approximations of the sensor readings although a limited number of nodes are compromised.
- Dynamic response to attack activities by the execution of a self-healing mechanism.

These properties should work together to provide accurate aggregation results securely without exhausting the network.

2.2 Requirements for Data Aggregation Security

Since WSNs share some properties with the traditional wireless networks, the data security requirements in the WSNs are similar to those in traditional networks (Perrig et al., 2002; Shi & Perrig, 2004). However, there are some unique specifications that can only be found in WSNs, as discussed in Section 1, which require more attention during the design process. This section discusses the security requirements for strengthening attack-resistant data aggregation protocols.

- **Data Confidentiality:** ensures that information content is never revealed to anyone unauthorized to receive it. It can be divided (in secure data aggregation schemes) into a hop-by-hop basis and an end-to-end basis. In the hop-by-hop basis, any aggregator point needs to decrypt the received encrypted data, apply some sort of

aggregation function, encrypt the aggregated data, and send it to the upper aggregator point. This kind of confidentiality implementation is not practical for the WSN since it requires extra computation, which leads to more delays in the network and increases the energy consumption. This kind of confidentiality also facilitates the adversary's mission. For example, the secrecy of sensed data is disclosed once any hop (or any sensor node included in the route) is compromised. On the other basis, the aggregator does not need to decrypt and encrypt the received data, and instead needs to apply the aggregation functions directly on the encrypted data by using homomorphic encryption (Westhoff et al., 2006). End-to-end confidentiality greatly reduces the energy consumption since there is no need for decryption and encryption at intermediate nodes. To the best of our knowledge, only SUM and AVE aggregation functions are implemented in the current literature.

- **Data Integrity:** ensures that the content of a message has not been altered, either maliciously or accidentally, during the transmission process. Confidentiality itself is not enough since an adversary is still able to change the data although it knows nothing about it. Suppose a secure data aggregation protocol provides only data confidentiality in order to defeat an adversary that is capable to compromise sensor nodes near aggregator points. The adversary can alter the sensed information to affect the overall aggregation results. Moreover, even without the existence of an adversary, data might be damaged or lost due to the nature of the wireless environment.

- **Data Freshness:** ensures that the data are recent and no old messages have been replayed, thereby protecting data aggregation protocols against replay attacks. In this kind of attack, it is not enough that these protocols provide only data confidentiality and data integrity because a passive adversary is able to listen to even encrypted messages, which is transmitted between sensor nodes, and can replay them later on to disrupt the data aggregation results. More importantly, the adversary can replay the distributed shared key and mislead the sensor about the current key used to secure sensing information and aggregated results.

- **Data Availability:** ensures that the network is alive and data are accessible. In the presence of malicious nodes, it is highly recommended that the network react to these bad (compromised) nodes and eliminate them. Once an attacker gets into the WSN by compromising a node, the attack can affect the network services and data availability, especially in those parts of the network where the attack has been launched. Moreover, the data aggregation security requirements should be carefully implemented to avoid extra energy consumption. If no more energy is left, the data will no longer be available. When the network size and the adversary capability are increased, it is preferable that a secure data aggregation protocol contains some of the following mechanisms to ensure a reasonable level of data availability in the network:

- **Self-healing** which can diagnose and react to the adversary's activities especially when it gets into the network, and then start corrective actions based on defined policies to recover the network or a node.
- **Aggregator rotation** that rotates the aggregation duties between honest nodes, to balance the energy consumption in the WSN.

- **Authentication:** allows the receiver to verify whether the message is sent by the claimed sender or not. The adversary will, therefore, not be able to participate and inject data into the network unless it has valid authentication keys. If the authentication is not implemented, the adversary can impersonate other nodes and get access to some sensitive data. In the aggregation context, without authentication, the adversary can masquerade the aggregator and report an aggregation result x' instead of x to the querier.

One major outcome of any secure data aggregation protocol is to provide the aggregated data as accurately as possible with a minimum number of bits transmitted within the network. A trade-off between data accuracy and the size of the aggregated data should be considered at the design stage. Before surveying secure data aggregation protocols, we discuss the security environments and the adversarial model considered in these protocols.

3. Adversarial Model

In this section, we describe the different capabilities that an adversary may have against the secure data aggregation protocols designed for WSNs. We further classify existing protocols according to the type of adversary the protocol designers considered.

3.1 Types of Attacks on Data Aggregation in Wireless Sensor Networks

WSNs are vulnerable to different types of attacks due to the nature of the transmission medium (broadcast), remote and hostile deployment location, and the lack of physical security in each node (Roosta et al., 2006). However, the damage caused by these attacks varies from one protocol to another, according to the adversarial model assumed by the protocol designers, which will be discussed in Section 5.3. The attacks that affect aggregation in WSNs are as follows:

- **Denial of Service Attack (DoS)** is a standard attack on WSNs that can be launched at any layer. One format of DoS attack can be radio signal transmission that interferes with the radio frequencies used by the WSN, which is sometimes called jamming. As the adversary capability increases, it can affect larger portions of the network. Another DoS format can include changing the node status from active to silent, thereby disabling the node. In the aggregation context, the DoS can be launched at the aggregator point in order to refuse executing aggregation functions and prevent data from travelling into the higher levels (or the base station).

- **Node Compromise Attack (NC)** is where the adversary is able to reach any deployed sensor node and extract the information stored on it. This attack is referred to as the supervision attack and sometimes the physical attack.

Considering the data aggregation scenario, once a node has been taken over, all the secret information stored on it can be extracted and the adversary can then participate in the aggregation activities.

- **Sybil Attack (SY)** is a type of attack where the attacker is able to present more than one identity within the network. It affects aggregation schemes in different ways. Firstly, an adversary may create multiple identities to generate additional votes in the aggregator election phase to make a malicious node the aggregator. Secondly, the aggregated result may be affected if the adversary is able to generate multiple entries with different readings. Thirdly, some protocols use witness-based mechanisms where witnesses are used to validate the aggregated data and the data is only valid if n out of m witnesses agreed on the aggregation results (Du et al., 2003). The adversary, however, can launch a Sybil attack and generate n or more witness identities to mislead the base station to accept incorrect aggregation results.

- **Selective Forwarding Attack (SF)** With no consideration about security, it is assumed in WSNs that each node will accurately forward received messages. A compromised node may refuse to do so since it is up to the adversary controlling the compromised node whether to forward the received messages or not. In the aggregation context, any intermediate nodes under the adversary supervision have the ability to launch the selective forwarding attack, and this subsequently affects the aggregation results.

- **Replay Attack (RE)** is a type of attack where the adversary is able to listen to the network and record some transmitted messages without even understanding their content and replays them later on. The adversary aims from launching this attack to mislead the aggregator with those old messages in order to affect the aggregation results.

Generally speaking, the adversary aims to inject false data into the network without revealing its existence. This happens when the adversary has the capability to launch any type of attack discussed above, or a mixture of them without revealing its existence. For example, the adversary can compromise a sensor node (NC attack) and subsequently generate more than one identity (SY attack) in order to affect the overall aggregation result. In a data aggregation scenario, the injected false value leads to a false aggregation result. A compromised node can report significantly biased or fictitious values, and perform a Sybil attack to affect the aggregation result.

3.2 Adversary Characteristics

Secure data aggregation protocols are threatened by two types of adversaries: passive and active adversaries. Differences between these two types are as follows:

- **Passive Adversary** is the adversary that takes advantage of the wireless communication nature (broadcasting) and eavesdrops on the traffic to obtain any important information about the sensed data. For example, if the adversary is able to hear the traffic near the aggregator point, it can gain some knowledge about the

aggregated result especially if the secure data aggregation scheme does not ensure data confidentiality service.

- **Active Adversary** is the adversary that interacts with the WSN by injecting packets, destroying nodes, compromising nodes, extracting sensitive data, and stopping/delaying packets from being delivered to the querier, etc. To put it another way, an active adversary can launch any type of attack listed in Section 3.1. The adversary has total access to the node's secrets, is able to extract *all* sensitive information stored in the sensor's memory and then harm the aggregation results.

As discussed in Section 2.1, there are three types of nodes in WSNs: sensor nodes, aggregators, and the base station with different functionalities and capabilities. The adversary's ability to compromise these three elements is discussed as follows:

- **Total Access:** The adversary that has total access to the network is powerful and has access to the whole WSN. Passive adversary can listen to all communications between. On the other hand, active adversary can interact maliciously with all types of components in the WSN (nodes, aggregators, base stations) by launching any type of attack listed in Section 3.1.

- **Partial Access:** This adversary has less power compared to the previous one. Its goal is to listen to communications between a subset of nodes in the network, if the adversary is passive. On the other hand, if the adversary is active, this means that it can only interact with a subset of nodes in the WSN.

3.3 Adversary Type

Adversaries in secure data aggregation protocols have two aspects: behavioural and network access. The adversary type can, therefore, be divided into four types:

- **Type 0:** refers to a passive adversary with limited access to the network. It eavesdrops on the communication in some parts of the network to which it has access. To the best of our knowledge, this type of adversary has never been considered in any secure data aggregation protocol.

- **Type I:** refers to a passive adversary that eavesdrops on the communication and is interested in revealing the encrypted data. The difference between type 0 and type I is the network access capability. Type I has total access to the network while type 0 has partial access.

- **Type II:** refers to an active adversary with limited access to the network (or it is able to compromise limited number of nodes) to launch attacks against secure data aggregation protocols and then mislead the base station about the aggregation results. Within its network limits, the adversary can launch any type of attacks listed in Section 3.1.

- **Type III:** refers to an active adversary that has total access to the network. It is interested in affecting the data aggregation results by launching any attack listed in Section 3.1 against any network component (nodes, aggregators, base stations).

We believe that this adversary classification can help to make better evaluation of the proposed schemes and facilitate making decisions on which protocol is more suitable for specific conditions as discussed in Section 5. In the following section, current secure data aggregation protocols are discussed in detail.

4. Current Secure Data Aggregation Protocols

To the best of our knowledge, there are four surveys in which current secure data aggregation protocols are compared. Setia et al. discussed the security vulnerabilities of data aggregation protocols and presented a survey of robust and secure data aggregation protocols that are resilient to false data injection attacks (2008). However, this survey covered only a few protocols. Sang et al. classified secure aggregation protocols into: hop-by-hop encrypted data aggregation and end-to-end encrypted data aggregation (2006). However, this classification does not detail the security analysis or the performance analysis of these protocols. Alzaid et al. classified these protocols based on how many times the data is aggregated during its travel to the base station, and whether these protocols have a verification phase or not (2008b). Their survey provided details on the security services offered by each protocol, security primitives used to defeat an adversary considered by the protocol designers. Ozdemir and Xiao surveyed the current work in the area of secure data aggregation and provided some details on the security services provided in each protocol (2009). We found that their security analysis is similar to Alzaid et al.'s work (Alzaid et al., 2008b).

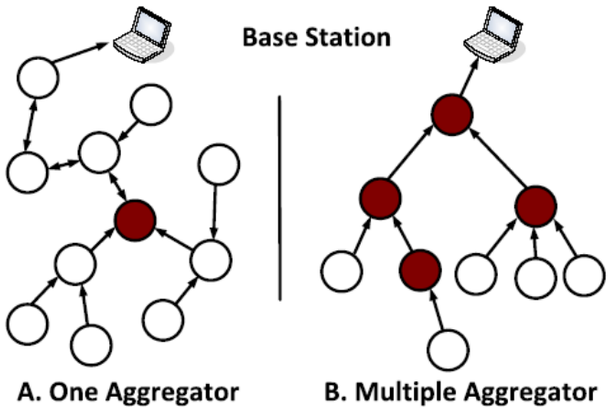


Fig. 2. Sketch of single and multiple aggregator models.

This section extends the work in (Alzaid et al., 2008b) and analyzes more secure data aggregation protocols, and then classifies them into two models: the one aggregator model and the multiple aggregator model (see Figure 2). Under each model, each secure data aggregation protocol either has a verification phase or does not, depending on security primitives used to strengthen the accuracy of the aggregation results although the protocol

is threatened by some malicious activities. To put in another way, this verification phase is used to validate the aggregation results (or the aggregator behaviour) by using methods such as interactive protocols between the base station (or the querier) and normal sensor nodes. We provide insights into the aggregation phase, verification phase, security primitives used to defeat the considered adversary, security services offered, and weaknesses of each protocol. Due to lack of space we discuss eight representative protocols in detail (four for each model) and summarize other protocols in subsections 4.1.5 and 4.2.5.

4.1 Single Aggregator Model

The aggregation process, in this model, takes place once between the sensing nodes and the base station or the querier. All individual collected physical phenomena (PP) in WSNs, therefore, travel to only one aggregator point in the network before reaching the querier. This aggregator node should be powerful enough to perform the expected high computation and communication. The main role of the data aggregation might not be fully satisfied since redundant data still travel in the network for a while until they reach the aggregator node, as shown in Figure 2-A. This model is useful when the network is small or when the querier is not in the same network. However, large networks are unsuitable places for implementing this model especially when data redundancy at the lower levels is high. Examples of secure data aggregation protocols that follow the one aggregator model are: Du et al.'s protocol (2003), Przydatek et al.'s protocol (2003), Mahimkar and Rappaport's protocol (2004), and Sanli et al.'s protocol (2004). These protocols are discussed in the following subsections.

4.1.1 Witness-based Approach for Data Fusion Assurance in WSNs (Du et al.)

4.1.1.1 Description

Du et al. proposed a witness-based approach for data fusion assurance in WSNs (2003). The protocol enhances the assurance of aggregation results reported to the base station. The protocol designers argued that selecting some nodes around the aggregator (as witnesses) to monitor the data aggregation results can help to assure the validity of the aggregation results.

The leaf nodes report their sensing information to aggregator nodes. The aggregator then needs to perform the aggregation function and forward the aggregation results to the base station. In order to prove the validity of the aggregation results, the aggregator node has to provide proofs from several witnesses. A witness is a node around the aggregator and also performs data aggregation like the aggregator node, but without forwarding its aggregation result to the base station. Instead, each witness computes the message authentication code (*MAC*) of the aggregation result and then sends it to the aggregator node. The aggregator subsequently must forward the proofs with its aggregation result to the base station.

4.1.1.2 Verification Phase

This protocol does not have a verification phase since the base station can verify the correctness of the aggregation results without the need to interact with the network. Instead, the protocol designers rely on the proofs that are computed by the witnesses and coupled with the aggregation results. Upon receiving the aggregation result with its proofs, the base station uses the n out of m +1 voting strategy to determine the correctness of the aggregation

results. In the *n* out of *m*+1 strategy, *m* denotes the number of witnesses nodes for each aggregator node while *n* denotes the minimum number of witnesses that should agree with the aggregation result provided by the aggregator. If less than *n* proofs agreed with aggregation result, the base station discards the result. Otherwise, the base station accepts the aggregation result.

4.1.1.3 Adversarial Model and Attack Resistance

The protocol designers considered an adversary that can compromise some aggregator nodes and witnesses as well. The designers, however, limited the adversary capability to compromising less than *n* witnesses for a single aggregator node. This type of adversary falls into the type II adversary, according to our discussion in Section 3.

From the discussion above, the NC attack is visible in this protocol. Once the adversary succeeds in NC attack against an aggregator node, it can then decide whether to forward the aggregation result and the proofs or not (SF attack). If the adversary keeps launching the SF attack, then one form of DoS attack is visible, too. The adversary, once it compromises an aggregator node, is able to replay an old aggregation result with its valid proofs instead of the current result to mislead the base station (RE attack). Finally, the adversary can launch NC attack against leaf nodes and then present multiple identities to affect the aggregation results (SY attack). The SY attack is visible in this protocol because the sensed PPs are not authenticated by the aggregator.

4.1.1.4 Security Primitives

The protocol designers used the *n* out of *m* +1 voting strategy to determine the correctness of aggregation results. This strategy is discussed in the verification phase for this protocol.

4.1.1.5 Security Services

The data aggregation security is provided by coupling the aggregation result with proofs from the witnesses around the aggregator node. These proofs, as discussed above, are *MACs* computed on the aggregation result to ensure its integrity and authenticate the witnesses to the base station. Other security services are not considered by the protocol designers.

4.1.1.6 Discussion

The security primitives used in this protocol to defend type II adversary is the *n* out of *m* + 1 voting strategy. This strategy authenticates witnesses and aggregators to the base station but not leaf nodes. The leaf nodes, therefore, are appropriate targets for the adversary to launch NC attack and then report invalid readings to aggregators. Moreover, the resource utilization in this protocol is poor for three reasons:

- The aggregator needs to receive *m* more proofs from the witnesses and the aggregator then needs to forward these extra proofs with its aggregation result.

- The number of times the aggregation takes place in the network is increased by *m* times, because every single aggregation function is repeated *m* times by the witnesses.

- Finally, the aggregation result with the proofs are travelled unchecked all the way to the base station, because the verification process is done at the base station.

4.1.2 Secure Information Aggregation in WSNs (Przydatek et al.)

4.1.2.1 Description

Przydatek et al. proposed a secure information aggregation protocol for WSNs which provides efficient sub-protocols for securely computing the median and the average of the measurements, estimating the network size and finding the minimum and the maximum sensor readings (2003). It consists of three types of network components: an off-site home server (or user), a base station (or aggregator), and a large number of sensors. The protocol designers claimed that their protocol provides resistance against stealthy attacks where the attacker's goal is to make the user accept false aggregation results without revealing its presence. We believe that stealthy attack can be accomplished by using any type of attack discussed in Section 3.1. The protocol employed, to achieve its goal, an aggregate-commit-prove approach where the aggregator performs aggregation activities and then proves to the user that it has computed the aggregation correctly. In this approach, the aggregator helps with computing the aggregation results and then forwards them to the home server together with a commitment to the collected data. The home server and the aggregator then use interactive proofs, where the home server will be able to verify the correctness of the results.

Due to lack of space, we limit our discussion to the MIN aggregation function. The designers proposed a secure MIN discovery sub-protocol that enables the home server (or the user) to find the minimum of the reported value. They, however, restricted the adversary capability: it can report only greater values than real values, not smaller. The sub-protocol works by first constructing a spanning tree such that the root of the tree holds the minimum element as illustrated in Algorithm 1.

The tree construction proceeds in iterations. Throughout the protocol, each sensor node S_i maintains a tuple of state variable (p_i , v_i , id_i), where p_i denotes the ID of the current parent of S_i in the tree being constructed, v_i denotes the smallest value seen so far, and id_i denotes the ID of the node whose value is equal to v_i . Each S_i initializes its state variables with its information as in steps 1, 2, and 3 in Algorithm 1. In each iteration, S_i broadcasts (v_i , id_i) to its neighbours. Let (v_i', id_i') denote a message sent by S' with a smaller value picked by S_i. Then, S_i updates its state by setting $p_i = S'$, $v_i = v_i'$, $id_i = id_i'$. The tree construction terminates after d iteration where d is an upper bound on the diameter of the network.

Algorithm 1 Finding the minimum value from nodes' sensed data

```
/*  code for sensor node i  */
/*  Initialization phase    */
1  p_i = S_i ;                  // current parent.
2  v_i = v_i ;                  // current sensed physical phenomenon.
3  id_i = S_i ;                 // owner of the current minimum value.
4  for i = 1 ... d  do
5      send (v_i , id_i) to all neighbours.
```

```
6    receive (v_j , id_j) from neighbors.
7    if (v_j < v_i) for sensor j then
8        p_i = S_j ;
9        v_i = v_j ;
10       id_i = id_j ;
11   end if;
12 end loop;
13 return < p_i , v_i, , id_i > ;
```

Upon constructing the tree, each node S_i authenticates its final state (p_i , v_i , id_i) using the key shared with the home server and then forwards it to the aggregator. The aggregator checks the consistency of the constructed tree with the values committed. If the check is successful, the aggregator commits to the list of all nodes and their states, finds the root of the constructed tree, and reports the root node to the home server. Otherwise, the aggregator reports the inconsistency. The commitment to the collected data is done using the Merkle hash tree (Merkle, 1980) to ensure that the aggregator used the data provided by sensors.

4.1.2.2 Verification Phase

The home server, upon receiving the aggregation results and the commitment of the collected data from the aggregator, needs to verify the correctness of the reported data. The home server checks whether or not the committed data is a good representative of the true values in the sensors network. This is done using interactive proofs, which is discussed in the security primitives' subsection a little later, where the home server checks if the aggregator is trying to provide an invalid aggregation result or not.

4.1.2.3 Adversarial Model and Attack Resistance

The protocol designers considered an adversary which can corrupt, at most, a small fraction of all the sensor nodes and then misbehave in any arbitrary way. However, more restrictions are put in their sub-protocols. They assumed that the adversary, in the secure MIN sub-protocol, cannot lie about its value or is uninterested in reporting a smaller value. This adversary falls in type II according to our discussion in Section 3.

According to the protocol designers, this type II adversary can launch NC attack but it is still unable to affect the secure MIN aggregation function, because the adversary is not allowed to report values smaller than the real values. We argue that this restriction should be relaxed because the adversary, with the ability to launch NC attack, can report whatever data it likes or selectively drop messages. We, thus, found that this protocol is non-resistant to SF attack.

Once the adversary decides to keep silent and stop reporting aggregation results, then one form of the DoS attack will be visible. Moreover, the protocol is protected against the RE attack due to the single usage of each temporary key shared with the base station. Finally, the protocol is protected against SY attack because the adversary cannot mislead the base

station to accept new hash chains for the faked identities in order to let them participate in the network.

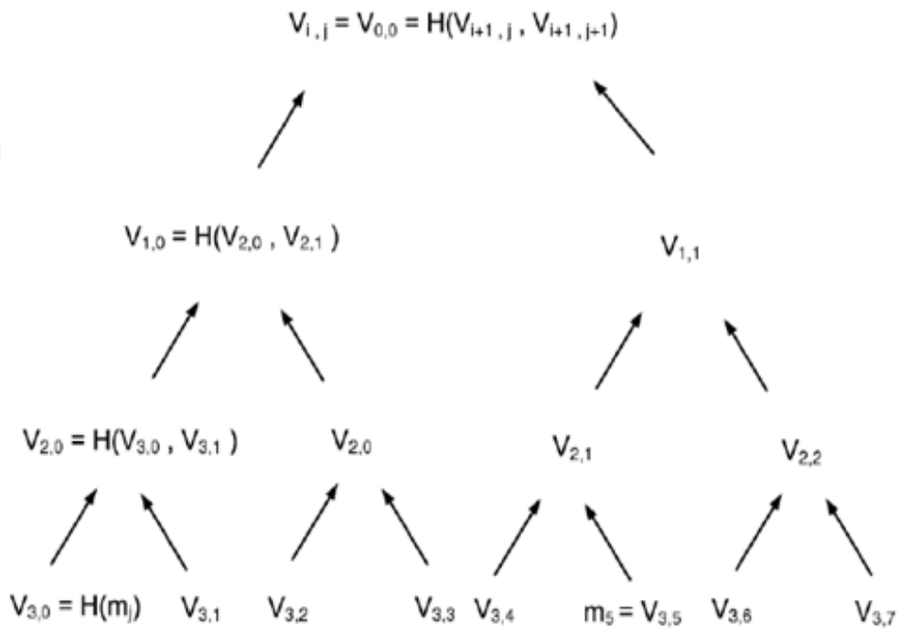


Fig. 3. An example of Merkle hash tree.

4.1.2.4 Security Primitives

The data aggregation security, in this protocol, is achieved by using the Merkle hash tree together with µTESLA (Perrig et al., 2002) and *MAC* security primitives. The aggregator constructs the Merkle hash tree over the sensor measurements $m_0, m_1, m_2, \ldots, m_7$ as in Figure 3, and then sends the root of the tree (called a commitment) to the home server. The home server can check whether the aggregator is cheating or not by using an interactive proof with the aggregator. It randomly picks a node in the committed list, say m_5 , and then traverses the path from the picked node to the root using the information provided by the aggregator. During the traversal, the home server checks the consistency of the constructed tree. If the checks are successful, then the home server accepts the aggregation result; otherwise, it rejects it. In other words, the aggregator sends the values of $v_{1,0}$, $v_{3,4}$, $v_{2,2}$ to the base station, and then the base station checks whether the following equality holds:

$$v_{0,0} = H(v_{1,0} \,||\, H(H(v_{3,4} \,||\, H(m_5)) \,||\, v_{2,2}))$$

4.1.2.5 Security Services

The protocol designers employed the Merkle hash tree together with µTESLA and *MAC* to defeat type II adversary. The usage of µTESLA and *MAC* provides authentication and data freshness to the network while the Merkle hash tree provides data integrity. Authentication is offered because only legitimate sensor nodes, with synchronized hash chains with the base station, are able to participate and contribute to the aggregation function. Data freshness is offered because of the single usage of the temporary key provided by µTESLA.

Unfortunately, data availability is not considered by the protocol designers due to the number of bits that travelled within the network in order to accomplish the aggregation task as discussed in Section 6.

4.1.2.6 Discussion

As discussed above, the protocol is able to check the validity of the aggregation result but with no further action to remove or isolate the node which caused inconsistency in the aggregation results. The authors also restricted the adversary capability: it can compromise the node but with no ability to report a value smaller than the real value when calculating the MIN aggregation function. We believe that this assumption should be relaxed because the adversary able to compromise nodes is able to perform whatever activities it likes. Once the assumption is relaxed, then the secure MIN sub-protocol should be revisited.

4.1.3 Secure Data Aggregation and Verification Protocol for WSNs (Mahimkar & Rappaport)

4.1.3.1 Description

A secure data aggregation and verification protocol is proposed by Mahimkar and Rappaport (2004). The protocol is similar to Przydatek et al.'s protocol, discussed in Section 4.1.2, except that it provides one more security service, which is data confidentiality. It uses digital signatures to provide data integrity service by signing the aggregation results.

This protocol is composed of two components: the key establishment phase and the secure data aggregation and verification phase. The key establishment phase generates a secret key for each cluster, and each node belonging to the cluster has a share of the secret key. The node uses this share to generate partial signatures on its reading. The second phase ensures that the base station does not accept invalid aggregation results from the cluster head (or the aggregator).

Each sensor node senses the required physical phenomenon (PP) and then encrypts it using its share of the cluster's private key. It then computes the *MAC* on its PP using the key shared between itself and the base station. The node after that sends these data (the encryption result and the *MAC*) to the cluster head which aggregates the nodes PPs and computes its average. The cluster head then broadcasts the average to all cluster members in order to let them compare their PPs with the average. If the difference is less than a threshold, the node (a cluster member) creates a partial signature on the average using its share of the cluster's private key and then sends it to the cluster head. The cluster head combines these signatures into a full signature and sends it along with the average value to the base station.

4.1.3.2 Verification Phase

The base station, upon receiving the average value and the full signature, verifies the validity of the signature using the cluster's public key. A valid signature is generated by a collusion of t or more nodes within the cluster. The base station accepts the aggregation result, which is the average value, once the signature validity is accepted. Otherwise, the base station rejects the aggregation result and uses the Merkle hash tree to ensure the integrity of the PPs. This is done in the same way suggested by Przydatek et al. and discussed in Section 4.1.2.

4.1.3.3 Adversarial Model and Attack Resistance

The protocol designers aimed to defeat an adversary that is able to compromise up to $t - 1$ nodes in each cluster, where t should be less than half of the total number of sensors in the cluster. This adversary falls into type II according to our discussion in Section 3. Type II adversary is able to launch NC attack as assumed by the designers of the protocol. Once the adversary compromised a sensor node, it can forward messages selectively to upper nodes or drop them (SF attack). Moreover, launching SF attack continuously makes one form of DoS attack visible in the network. The adversary can further replay an old message with its own valid signature, instead of the current message, to affect the aggregation results. Finally, the protocol is SY attack resistant since each node should have a legitimate share of the cluster's private key that cannot be generated by the adversary.

4.1.3.4 Security Primitives

To defeat the adversary considered in this protocol, the designers used Merkle hash tree together with encryption and digital signature. They used elliptic curve cryptography to encrypt PPs reported to the cluster head, digital signature concept to sign aggregation results, and the Merkle hash tree to verify the integrity of the reported aggregation results once the signature verification failed. The encryption and digital signature are common concepts in the security domain and thus discussion about them is out of the chapter's scope. The Merkle hash tree, however, is within the scope of this chapter and already discussed in Section 4.1.2.

4.1.3.5 Security Services

The protocol, through the key establishment component, provides authentication service because only the cluster members with legitimate shares are able to participate in the aggregation processing. Data confidentiality and integrity are offered through the aggregation and verification component. Elliptic curve encryption provides data confidentiality while digital signatures and the Merkle hash tree enhance data integrity of the aggregation results. Data freshness, however, is not considered by the protocol designers.

4.1.3.6 Discussion

If the adversary compromised any sensor node except the aggregator, it is able to affect the aggregation result by reporting invalid PPs. Wagner proved that the average function, which is implemented in this protocol as the aggregation function, is insecure in the existence of only one compromised sensor node (Wagner, 2004). Even worse; when the adversary succeeds in compromising the cluster head (or the aggregator), the adversary can then replay old but valid signed aggregation results to mislead the base station.

Moreover, the protocol designers considered only the average function and, replacing this function with other functions is impossible given the same protocol run. In the current scenario, each sensor node is able to check the aggregation result by dividing its PP by the number of sensor nodes in its cluster, and then comparing the result with the average value broadcasted by the cluster head. The sum function, for example, cannot be implemented because each sensor node encrypts its PP using a different share of the cluster private key.

4.1.4 Secure Reference-based Data Aggregation Protocol for WSNs (Sanli et al.)

4.1.4.1 Description

Sanli et al. proposed a secure reference-based data aggregation protocol that encrypts the aggregation results and applies variable security strength at different levels of the cluster heads (or aggregators) hierarchy (2004). The differential data, which is the difference between the reference value and the sensed data, is reported to aggregator points instead of the sensed data itself in order to reduce the number of transmitted bits.

The protocol designers argued that intercepting messages transmitted at higher levels of clustering hierarchy provides a summary of a large number of transmissions at lower levels. The designers, therefore, believed that the security level of the network should be gradually increased as messages are transmitted through higher levels. Based on this observation, they chose a cryptographic algorithm that allows adjustment of its parameter and the number of encryption rounds to change its security strength as required.

Instead of sending the raw data to the aggregator, a sensor node compares its sensed data with the reference data and then sends the encryption of the difference data. The reference data is taken as the average value of a number of previous sensor readings, N, where $N \geq 1$. The aggregator, upon receiving these differential data, performs the following activities:

- Decrypts the data and then determines the distance to the base station in the number of hops (h).
- Encrypts the aggregation result using RC6 with the number of rounds calculated as:

$$number\ of\ rounds = \frac{1}{h} * 100 \quad (1)$$

Forwards the encrypted aggregated data to the base station.

4.1.4.2 Verification Phase

This protocol does not contain a verification phase to check the validity of the aggregation results. The protocol designers, instead, relied on the security primitives, RC6, to enhance the security for the aggregation results. The protocol is designed to encrypt the aggregation results with different numbers of encryption rounds, depending on how far the aggregator node is from the base station. Once the base station has received the encrypted aggregation results, it decrypts them with the corresponding keys.

4.1.4.3 Adversarial Model and Attack Resistance

The protocol designers did not discuss the adversary capability that was considered in their protocol. We believe, from their discussion in the paper, that the adversary type falls into the category of type I adversary for the following reasons:

- They relied only on encryption to provide accurate data aggregation.
- A single node compromise can breach the security of the protocol. For example, once the adversary compromised an aggregator node, the privacy and accuracy of the aggregation results can be manipulated and then affect the overall aggregation activities of the system.

4.1.4.4 Security Primitives

To defeat type I adversary, the designers of the protocol used the block cipher RC6. They adjust the number of rounds, which RC6 performs to accomplish an encryption operation, depending on how far the aggregator point is from the base station. The closer the aggregator is, the larger the number of rounds should be used.

4.1.4.5 Security Services

The data aggregation security is achieved by encrypting travelled data using the block cipher RC6. This provides a data confidentiality service to the network. Data freshness is also provided due to the key update component adhered to the aggregation component. Other security services are not considered because of the type of adversary considered by the protocol designers.

4.1.4.6 Discussion

The security primitives, used to defeat the type I adversary, is impractical for use in constrained devices such as sensor nodes. Law et al. constructed an evaluation framework in which suitable block cipher candidates for WSNs can be identified (2006). They concluded, based on the evaluation results, that RC6 is lacking in energy efficiency (i.e., a large RAM consumer), and performs poorly on 8/16 bits architectures. They further concluded that RC6 with 20 rounds is secure against a list of attacks such as chosen ciphertext attack. However, the number of rounds for RC6 encryption in Sanli et al.'s protocol can be as low as 10 rounds once the aggregator node is 10 hops away from the base station, according to equation 1.

4.1.5 Other Protocols

Wagner proposed a mathematical framework for evaluating the security of several resilient aggregation techniques/functions (2004). The paper measures how much damage an adversary can cause by compromising a number of nodes and then using them to inject erroneous data. Wagner described a number of better methods for securing the data aggregation such as how the median function is a good way to summarise statistics. However, this work focused only on examining the security of the aggregation functions at the base station without studying how the raw data are aggregated. Furthermore, Wagner claimed that trimming and truncation can be used to strengthen the security of many aggregation primitives by eliminating possible outliers. However, eliminating abnormal data with no further reasoning is impractical in some applications such as monitoring bush-fire.

4.2 Multiple Aggregator Model

In this model, collected data in WSNs are aggregated more than once before reaching the final destination (or the querier). This model achieves greater reduction in the number of bits transmitted within the network, especially in large WSNs, as illustrated in Figure 1. The importance of this model is growing as the network size is getting bigger, especially when data redundancy at the lower levels is high. A sketch of the multiple aggregator model can be found in Figure 2-B. Examples for secure data aggregation protocols that fall under this model are: Hu and Evans's protocol (2003), Jadia and Mathuria's protocol (2004), Westhoff

et al.'s protocol (2006), and Sanli et al.'s protocol (2004). These protocols are discussed in the following subsections.

4.2.1 Secure Data Aggregation for Wireless Networks (Hu & Evans)

4.2.1.1 Description

Hu and Evans proposed a secure aggregation protocol that achieves resilience against node compromise by delaying the aggregation and authentication at the upper levels (2003). The required physical phenomena (PP) are, therefore, forwarded unchanged and then aggregated at the second hop instead of aggregating them at the immediate next hop. Thus, the parents need to buffer the data to authenticate it once the shared key is revealed by the base station. It is the first attempt towards studying the problem of data aggregation in WSNs once a node is compromised.

Each sensor node shares a temporary symmetric key with the base station, which lasts for a single aggregation calculation. The base station periodically broadcasts these authentication keys as soon as it receives the aggregation result. Each leaf node, as a part of the aggregation phase, transmits its PP to its parent. This transmission includes the node ID, the sensed PP, and the message authentication code $MAC_{K_{ID}}(ID, PP)$. It uses the temporary key shared with the base station, but not yet known to the other nodes, to calculate the *MAC*. The parent (or any intermediate node) applies the aggregation function on messages received from its children, then calculates the *MAC* of the aggregation result, and transmits messages and *MACs* received from its direct children along with the *MAC* computed on the aggregation result. The parent, which has grandchildren, is permitted to remove its grandchildren's raw data (or PPs) and confirm the aggregation result done by its children (or parents of its grandchildren). It is important that each parent stores raw data received from its children (and its grandchildren if it available) and the *MAC* computed on the reported data from its children (and its grandchildren if available). The parent will use this information at the end of the aggregation process when the base station reveals the temporary keys, as discussed in the following subsection.

4.2.1.2 Verification Phase

This protocol has a verification phase where the base station interacts with sensor nodes and aggregators in order to verify the aggregation results. The protocol designers used μTESLA protocol, which is discussed in the security primitives' subsection, to achieve the interaction between the base station and sensor nodes. When aggregation results arrive at the base station, the base station reveals the temporary symmetric keys shared with every node. Every parent is now able to verify whether the information (raw data and the *MAC*) stored for its children is matched or not. If the parent detects an inconsistent *MAC* from a child or a grandchild, it sends out an alarm message to the base station along with *MAC* computed using the node's temporary key.

4.2.1.3 Adversarial Model and Attack Resistance

The most serious threat considered by the designers of the protocol is that an adversary that can compromise the network to provide false readings without being detected by the operator. Each intermediate node (parent) can thus modify, forge, discard messages, or transmit false aggregation values. The designers, however, limited the adversary capability

to not launching an NC attack for two consecutive nodes in the hierarchy. This type of adversary falls into type II according to our discussion in Section 3.
SY and RE attacks, in this protocol, are not visible while DoS, NC, and SF are visible. The adversary considered by the designers is able to compromise any sensor node (either a leaf node or an aggregator) - this is the NC attack. Once an intermediate node is compromised, the adversary is easily able to launch the SF attack. Even worse, the adversary can decide to keep silent and stop reporting aggregation results, which is one form of the DoS attack. The protocol, however, is protected against the RE attack due to the single usage of each temporary key shared with the base station. Finally, the protocol is protected against SY attack because the adversary cannot mislead the base station to accept new hash chains for the faked identities.

4.2.1.4 Security Primitives

In this protocol, *MAC* and μTESLA are used to provide authentication, data integrity, and data freshness. *MAC* is a well known technique in the cryptographic domain used to ensure authenticity and to prove the integrity of the data. It is calculated using a key shared between two parties (the sender and the receiver). These keys are updated by using μTESLA protocol that delays the disclosure of symmetric keys to achieve asymmetry (Perrig et al., 2002). The base station generates the one-way key chain of length n. It chooses the last key K_n and generates the remaining values by applying a one-way function F as follows:

$$K_j = F(K_{j+1})$$

Because F is a one-way function, anybody can compute backward, such as compute $K_0, K_1, \ldots, K_j$ given K_{j+1} , but nobody can compute forward such as compute K_{j+1} given $K_0, K_1, \ldots, K_j$. In the time interval t, the sender is given the key of the current interval K_t by the base station through a secure channel, and then the sender uses the key to calculate MAC_{K_t} on its PP in that interval. The base station then discloses K_t after a delay and then other nodes will be able to verify the received MAC_{K_t} .

4.2.1.5 Security Services

The protocol designers regarded data confidentiality of messages to be unnecessary for their protocol. They focused only on the integrity of aggregation results by using μTESLA protocol, which also provides authentication and data freshness services. Authentication is offered because only legitimate sensor nodes, with synchronized hash chains with the base station, are able to participate and contribute to the aggregation function while data freshness is offered because of the single usage of the temporary key. Unfortunately, data availability is not considered by the designers because each parent has to store and verify received information from its children and grandchildren. This verification requires each parent to listen to every key revealed by the base station until it hears the keys of its children and grandchildren. Even worse for data availability, the data keeps travelling towards the base station even when it has been corrupted because the keys are revealed when the aggregation results reach the base station. Another factor that affects data availability is, once a compromised node is detected, no practical action is taken to reduce the damage

caused by this compromise, and the compromised node can still participate in the aggregation activities.

4.2.1.6 Discussion

The protocol designers considered data integrity and used μTESLA to defeat type II adversary. The protocol is able to detect a single node compromise, but without further action to remove or isolate this compromised node. Much worse, once a grandfather node detects a node compromise, it could not decide whether the cheating node is its child or grandchild. The protocol, moreover, fails to provide data integrity once the adversary compromised two consecutive nodes successfully in the hierarchy such as the parent and the grandparent. The protocol also suffers from extra memory overhead because of the delayed authentication and the need to buffer the data received by parents to be authenticated later. Finally, parents waste some energy listening to some of the revealed keys that are not intended for them.

4.2.2 Efficient Secure Aggregation in Sensor Networks (Jadia & Mathuria)

4.2.2.1 Description

Hu and Evans in their protocol, discussed in Section 4.2.1, did not consider data confidentiality service. Jadia and Mathuria, however, argued that messages relayed in data aggregation hierarchy may need confidentiality. Thus, they proposed a secure data aggregation protocol in WSNs that enhances the security services provided by Hu and Evans's protocol by adding data confidentiality (Jadia & Mathuria, 2004). This protocol uses encryption for confidentiality but without requiring decryption at intermediate nodes. The designers of the protocol adopted an encryption method where the data is added to a sufficiently long random encryption key. Let K_A denote the master key shared between node A and the base station. The encryption of the sensed PP reported by a sensor node A can be calculated as follows:

$$C_{K_A} = (PP_A + K_A)$$

After encrypting the required *PPs*, node *A* computes two *MACs* on these *PPs*. One *MAC* is calculated by using one-hop pairwise key shared with the node's parent while the second *MAC* is calculated using two-hop key shared with the node's grandparent. The aggregation phase is accomplished in the same way as the Hu and Evans's protocol, except for two differences listed below:

- Leaf nodes encrypt the node's *PPs* before sending them.
- Leaf nodes compute two *MACs* on the encrypted data.

The leaf node then forwards its *ID*, encrypted data, and two *MACs* to its parent. The parent node (say node *C*) receives the message and verifies the origin of the data using the one-hop pairwise shared key. It performs the aggregation over the encrypted data but does not transmit this aggregated value. The aggregation calculation is performed on the encrypted data received from its children (node *A* and node *B*) as follows:

$$Encrypted\ Aggregation\ Result\ (EAR) = C_{K_A} + C_{K_B} + C_{K_C} \quad (2)$$

Node C then calculates a MAC of EAR using the two-hop pairwise key shared with its grandparent node, and transmits it along with the encrypted *PPs* and *MAC*s received from its children (of course without the MAC intended for itself).

4.2.2.2 Verification Phase

This protocol does not have a verification phase. The protocol designers argued that the two *MACs*, which are discussed in Section 4.2.2.1, help to provide the integrity of the data while minimizing the communication required between the base station and sensor nodes. In other words, the verification phase in Hu and Evans's protocol, where the base station reveals temporary shared keys with nodes, is replaced with the pairwise-based *MACs* in order to improve data availability in the network. The designers, however, did not discuss how these pairwise keys are distributed and how much bandwidth and energy consumption are required.

If the base station did not receive alarm messages from parents regarding inconsistency between encrypted data and *MACs* computed on them, the base station decrypts the aggregation result (EAR) from equation 2 as follows:

$$Aggregation\ result = EAR - (K_A + K_B + K_C)$$

4.2.2.3 Adversarial Model and Attack Resistance

Since this protocol is an extension to Hu and Evans's protocol discussed in Section 4.2.1, the protocol designers considered a similar adversary type that falls into type II adversary according to our discussion in Section 3.

Moreover, DoS, NC, and SF attacks are visible in this protocol due to the capability of type II adversary and to the same discussion that is given in Section 4.2.1.3. The protocol is SY and RE resistant due to the design assumption that the authentication and encryption keys are changed with every message. However, no details on changing these keys are given.

4.2.2.4 Security Primitives

The protocol designers employed MAC together with encryption to defeat type II adversary. They used pairwise keys to calculate the MAC and the concept of privacy homomorphic encryption to perform aggregation on the encrypted data, as discussed in Section 4.2.2.1.

4.2.2.5 Security Services

This protocol provides data confidentiality, data integrity, data freshness, and authentication services. The usage of two *MACs*, which are calculated by one-hop and two-hop pairwise keys, provides data integrity and authentication for the aggregation results. Data confidentiality is provided by using the adopted end-to-end encryption that is discussed in Section 4.2.2.1. Finally, data freshness service is visible in the network due to the designers' assumption that the authentication and encryption keys are changed with every message.

4.2.2.6 Discussion

As discussed above, the designers of the protocol added data confidentiality service to security services provided by Hu and Evans's protocol. The protocol, here, suffers from the same weaknesses that Hu and Evans's protocol suffered from, discussed in Section 4.2.1.6. However, the memory overhead weakness is not visible in this protocol because it uses pairwise keys and does not need to keep copies of *MACs* information until the base station reveals temporary keys.

4.2.3 Concealed Data Aggregation for Reverse Multicast Traffic in WSNs (Westhoff et al.)

4.2.3.1 Description

Westhoff et al. solved the problem of aggregating encrypted data in WSNs, and proposed a secure data aggregation protocol that provides aggregator nodes with the possibility to perform aggregation functions directly on ciphertexts (2006). This work is an extension to their initial work in (Girao et al., 2005). It uses an additive and multiplicative Privacy Homomorphic (PH) encryption scheme (Domingo-Ferrer, 2002) in order to provide end-to-end encryption. The aggregator nodes do not need to decrypt encrypted messages when they aggregate them. If the usual encryption algorithms, such as RC5, were used instead of PH to provide data confidentiality, hop-to-hop encryption then should be used instead of end-to-end encryption. This is because usual algorithms do not let aggregator nodes apply aggregation functions directly on ciphertexts. Hop-by-hop encryption means that every intermediate node has to decrypt received encrypted messages, and then aggregate them according to the corresponding aggregation function, encrypt the aggregation results, and finally forward the aggregation results to upper nodes. Westhoff et al.'s protocol employs the Domingo-Ferrer's encryption function that chooses the ciphertext corresponding to given plaintexts (or messages) from a set of possible ciphertexts. The public parameters, for the encryption function, are a positive integer $d \geq 2$, and a large integer g that has many small divisors. There should be, at the same time, many integers $< g$ that can inverted modulo g. The secret key is computed as:

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

The plaintext $r \epsilon \mathbb{Z}_{g'}$ is chosen such that $r^{-1} \, mod \, g$ exists, where $log_{g'} \, g$ indicates the security level provided by the function. The set of plaintext is $\mathbb{Z}_{g'}$ and the set of ciphertext is $(\mathbb{Z}_g)^d$. The encryption process is executed at leaf nodes as follows:

- Randomly split the plaintext $a \epsilon \mathbb{Z}_{g'}$ into secretes $a_1, a_2, \ldots, a_d$ such that

$$\sum_{j=1}^{d} \left(a_j \, mod \, g'\right) = \, a$$

- Compute $E_k(a) = (a_1 r^1 \, mod \, g \, , \, a_2 r^2 \, mod \, g \, , \ldots , a_d r^d \, mod \, g)$

Leaf nodes then forward the encrypted data to aggregator nodes where PH is used to apply aggregation function on these encrypted data with no need to decrypt them. The decryption

process is performed at the base station (or the querier) and is discussed when we describe the verification process in the following subsection.

4.2.3.2 Verification Phase

This protocol does not have a verification phase. The designers of the protocol, instead, relied on the security primitive, discussed in Section 4.2.3.4, to defeat the considered type of adversary. The protocol is designed to encrypt the required physical phenomenon in a way that aggregators are able to apply aggregation functions directly on ciphertexts. The aggregators then forward the aggregation results to upper nodes. When these aggregation results reach the querier, the querier decrypts them as follows:

- Compute the j^{th} coordinate by $r^{-j}\ mod\ g$ to retrieve $a_j\ mod\ g$.
- In order to compute a, the querier computes $D_k\big(E_k(a)\big) = \sum_{j=1}^{d}(a_j\ mod\ g')$

4.2.3.3 Adversarial Model and Attack Resistance

The designers of the protocol aimed to defeat passive adversaries that eavesdrop on communication between sensor nodes, aggregators, and the base station. However, the designers extended the capability of the adversary to be able to takeover aggregator nodes but not other network components. Thus, we classify this adversary to fall under type II category due to its capability to launch NC attack.

Since the adversary is able to compromise aggregator nodes, it can then launch RE attack by replacing old but valid encrypted messages as long as encryption keys of leaf nodes have not been updated/renewed. Once an aggregator is compromised, the adversary is easily able to launch SF attack. Even worse, the adversary can decide to keep silent and stop reporting aggregation results, which is one form of the DoS attack.

4.2.3.4 Security Primitives

The protocol designers employed Privacy Homomorphism (PH) to defeat the type II adversary. During the last few years, PH encryption schemes have been studied extensively since they proved to be useful in many cryptographic applications such as electronic elections (Grigoriev & Ponomarenko, 2003), sensor networks (Castelluccia et al., 2005; Westhoff et al., 2006) and so on. Homomorphic cryptosystem is a cryptosystem that allows direct computation on encrypted data by using an efficient scheme. It is an important tool that can be used in a secure aggregation scheme to provide end-to-end privacy if needed.

The classical RSA scheme is a good example of a deterministic, multiplicative homomorphic cryptosystem on $M = \frac{\mathbb{Z}}{N\mathbb{Z}}$, where N is the product of two large primes (Rivest et al., 1978). Let K_e, K_d, E, D, m, c denote the private key, public key, encryption function, decryption function, message in plaintext, ciphertext, respectively. Thus, $C = \frac{\mathbb{Z}}{N\mathbb{Z}}$ is the ciphertext space while the key space is:

$$K = \{(k_e, k_d) = ((N, e), d) | N = pq, ed \equiv 1\ mod\ \varphi(N)\}$$

The encryption of any message $m \epsilon M$ is defined as:

$$E_{k_e}(m) = m^e \ mod \ N$$

while the decryption of any ciphertext $c \ \epsilon \ C$ is defined as:

$$D_{k_e\,,\,k_d}(c) = c^d \ mod \ N = m \ mod \ N$$

Obviously, the encryption of the product of two messages $m_1, m_2 \epsilon \ M$ can be computed by multiplying the corresponding ciphertexts:

$$\begin{aligned} E_{k_e}(m_1 \ \odot \ m_2) &= (m_1 \ m_2)^e \ mod \ N \\ &= (\ m_1^e \ mod \ N)(\ m_2^e \ mod \ N) \\ &= E_{k_e}(m_1) \ \odot \ E_{k_e}(m_2) \end{aligned}$$

4.2.3.5 Security Services

The data aggregation security is provided by encrypting the reported data and thus only data confidentiality is provided. Other security services, discussed in Section 2.2, are not provided due to the focus of the paper.

4.2.3.6 Discussion

The security primitive used to defeat the type II adversary is PH. This primitive is impractical to be used in constraint devices, such as the sensor node, due to its high computational cost (Westhoff et al., 2006). The protocol designers argued that their protocol considered this disadvantage, the high computational cost, by rotating the aggregation duties between aggregators to balance the energy consumption.

Moreover, Wagner proved that PH is insecure against chosen plain text attacks (Wagner, 2003). The protocol designers argued that for data aggregation scenarios in WSNs, the security level is still adequate and they used this encryption transformation as a reference PH.

Unfortunately, this protocol can support only "average" and "movement detection" aggregation functions. Applying PH on the context of WSNs in order to support other aggregation functions is an open area of research.

4.2.4 Secure Reference-based Data Aggregation Protocol for WSNs (Yang et al.)

4.2.4.1 Description

Yang et al. proposed a secure data aggregation for WSNs that can tolerate more than one node compromise (2006). The protocol is composed of two components: divide-and-conquer and commit-and-attest. In the former, the protocol uses a probabilistic grouping technique that partitions nodes in a tree topology into several logical groups. In the latter, a commitment-based hop-by-hop aggregation is performed in each group to generate a group aggregate. The base station then identifies the suspicious groups based on a set of group aggregates. Each group under suspect participates in an attestation process to prove the validity of its group aggregation result.

A leaf node encrypts its *ID*, physical phenomenon (*PP*), count value (*C*), and the query sequence number (*SQ*) using a pairwise key shared with its parent. The count value represents the number of the node's children, and therefore *C* for any leaf node is always

zero. It then forwards to its parent the encryption result, a *MAC* computed on inputs to the encryption function, and one bit aggregation flag. This flag instructs the node's parent upon receiving the transmission whether there is a need for further aggregation (flag=0) or not. When an intermediate node receives a message from its child, it first checks the flag and then follows one of the following scenarios:

- **1st scenario (flag=1):** the intermediate node forwards the packet untouched to the base station via its parent.
- **2nd scenario (flag=0):** the intermediate node decrypts the received message and then checks whether or not the received data is a response to the current query. Once this checking is passed, the intermediate node adds its own *PP* and other aggregation results received from other children (with flag=0) to the received data. The *C* is subsequently updated by adding up count values of all other participants.

To set the aggregation flag to one (no more aggregation) for this intermediate node, the node performs the following check:

$$H(SQ|ID) < F_g(C) \tag{3}$$

where *H* is a secure pseudo random function that uniformly maps the input values into the range of [0,1] and F_g is a grouping function that outputs a real number between [0,1]. This check helps the intermediate node to decide whether it is a leader node or not. Using the pairwise key shared with its parent, non-leader node encrypts its *ID*, new *C*, aggregation result, and *SQ*. It then sets the flag to zero and forwards these data along with a *MAC*, which is computed on inputs to the encryption function, and an XOR result for all *MACs* received from its children and included in this aggregation. The leader node on the other hand performs the same operation as the non-leader node, except that it encrypts the new aggregation using the key shared with the base station and sets the flag to one.

4.2.4.2 Verification Phase

The base station, upon receiving the aggregation result from a leader node, needs to verify whether the received aggregation result is accurate and came from a genuine leader node. It decrypts this aggregation result and then applies equation 3 to check the legitimacy of the node as a leader node. Once the test is passed, the base station needs to check the validity of the received aggregation result. First, the base station uses an adaptive Grubbs test (Grubbs, 1969) to verify the abnormality in the aggregation result before accepting or rejecting the received aggregation result. The base station then attests the group where the abnormal aggregation result is reported. Details on checking the validity of the aggregation result is given in the security primitives' section later.

4.2.4.3 Adversarial Model and Attack Resistance

The protocol designers considered an adversary that can compromise a small fraction of sensor nodes to obtain the keys as well as reprogramming these sensor nodes with attacking code. This type of adversary falls within the type II according to our discussion in Section 3.

Although the protocol designers mentioned that they did not consider any type of behaviour-based attack such as SF and DoS attacks, their protocol is examined against these attacks for the sake of a complete survey. We argue that if the adversary is able to launch NC attack in order to mislead the base station about the aggregation results, the adversary can also perform the activity of the SF attack for the same purpose. Beside the visibility of NC and SF attacks, the DoS attack is visible in the network, too. An example of the visibility of the DoS attack is similar to what was discussed in Section 4.2.1.3. The protocol, however, is RE and SY attack-resistant due to the query sequence number embedded in the reported PPs and to the pairwise key updates, respectively.

4.2.4.4 Security Primitives

The designers of the protocol used an encryption algorithm, μTESLA, adaptive Grubbs test, and attestation mechanism to defeat the type II adversary. Since the designers did not provide details about the encryption algorithm and μTESLA was discussed in Section 4.2.1, the adaptive Grubbs test and the attestation mechanism are discussed here.

The adaptive Grubbs test, as shown in Algorithm 2, first computes the sample statistic for each datum X in the set by $\frac{X-\mu}{s}$, where m and s are the mean and the standard deviation of the data, respectively. The result represents the datum's absolute deviation from the mean in units of the standard deviation. To decide whether H_0 should be accepted or not, the test compares the p-value computed based on the sample statistic with the predefined significance level α (α = 0 typically), where p-value is set as the product of the p-values of the data aggregation and the count (the number of participants in the aggregation). When the p-value is smaller than α, H_0 is rejected and the datum under consideration is an outlier, and then the attestation mechanism is called.

The attestation process is similar to the Merkle hash tree discussed in Section 4.1.2. The base station interacts with the group under suspect to prove the correctness of its group aggregation result.

Algorithm 2 Grubbs test algorithm

Input: a set T of n tuple (x, c_x, Agg_x), where x is group leader ID, c_x is group count value, Agg_x is group aggregation result, and n is the total number of groups;

Output: a set L of leader *IDs* of groups with invalid aggregation results.

Procedure:

1 loop

2 compute μ_c and s_c for all counts in set T;

3 compute μ_v and s_v for all values in set T;

4 find the maximum count value c_x in set T;

5 compute statistic Z_c for count c_x as $\frac{|c_x-\mu_c|}{s_c}$;

6 compute p-value P_c based on the statistic Z_c;

7 compute statistic Z_v for corresponding values Agg_x as $\frac{|Agg_x-\mu_v|}{s_c}$;

8 compute p-value P_v based on the statistic Z_v;

9 **if** $(P_c * P_v) < \alpha$ **then**

10 $T = T - \{(x,\ c_x\ , Agg_x)\}$;
11 $L = L \cup \{x\}$;
12 **else**
13 break;
14 **end if**;
15 end loop
16 return L;

4.2.4.5 Security Services

The data aggregation security is achieved by encrypting PP*s* destined to the base station and then by checking the validity of the aggregation results. This ensures data confidentiality, authentication, and data integrity within the network. Due to the query sequence number, which is embedded in any response, data freshness is offered, too. Data availability, however, is not visible because of the high number of transmission required to accomplish the aggregation activities. More details are given in Section 6.

4.2.4.6 Discussion

As discussed above, the protocol designers used an adaptive test to check the validity of aggregation results. This adaptive test is subject to attack when some nodes are compromised. The test uses reported aggregation results to compute the μ and *s* (see Algorithm 2). Compromised nodes can collude and report invalid aggregation results to mislead the calculation of the mean of the data (*m)* and then affect steps 3-16 in Algorithm 2. This will affect the base station's decision and may enforce it to start the attestation process with honest groups instead of malicious groups. Moreover, invalid aggregation results are attested (or verified) through centralized verification that incurs high communication cost.

4.2.5 Other Protocols

Furthermore, an extension to Westhoff et al.'s protocol is proposed by Castelluccia et al. (2005). It uses a modular addition instead of the XOR (Exclusive-OR) operation found in the stream ciphers. Thus, even if an aggregator is compromised, original messages cannot be revealed by an adversary (assuming that the aggregator does not have the encryption key). The authors claimed that the privacy protection provided by this protocol is comparable to the privacy protection provided by a protocol that performs end-to-end encryption with no aggregation. However, they admit that their proposed scheme generates significant overhead if the network is unreliable since sensors' identities of non-responding nodes must be sent together with the aggregated result to the base station. More importantly, this scheme provides only one security property which is data confidentiality.

Chan et al. extended Przydatek et al.'s protocol by applying the aggregate-commit-prove framework in a fully distributed network instead of single aggregator model (2006). The protocol detects the existence of any misbehaviour in the aggregation phase. The protocol designers, however, did not consider data availability because they did not aim either to identify or remove nodes that caused this misbehaviour. In general, their protocol offers the same as Przydatek et al.'s protocol: authenticity, data integrity, and data freshness. Each

parent performs an aggregation function whenever it has heard from its child nodes. In addition, it has to create a commitment to the set of the input used to compute the aggregated result by using the Merkle hash tree. It then forwards the aggregated data and the commitment to its parent until it reaches the base station. Once the base station has received the final commitment values, it rebroadcasts them into the rest of the network in an authenticated broadcast. Each node is responsible for checking whether its contribution was added to the aggregated result or not. Once its reading is added, it sends an authentication code to the base station where the authentication code for node R is $MAC_{K_R}(N \,||\, OK)$, where K_R is the key that node R shares with the base station, and N denotes a nonce. For communication efficiency, the authentication codes are aggregated along the way to the base station. However, one missing authentication code for any reason leads the base station to reject the aggregated result. Furthermore, noticeable delay, too much transmission, and computation are added as consequences of adding security to this protocol.

Frikken and Dougherty improved the performance of Chan et al.'s protocol by proposing a new commitment tree structure (2008). Let Δ denote the degree of the aggregation tree and n denote the number of sensor nodes. They claimed that their protocol requires each node to perform $\mathcal{O}(\Delta\ log\ n)$ communication while Chan et al.'s protocol requires $(\Delta\ log^2\ n)$.

Most secure data aggregation, discussed previously, can detect the manipulations of aggregation results and then reject it. They have no further attempts to identify nodes which caused the manipulations, and thus a single node compromise gives the adversary the ability to disturb the network resources by participating maliciously during the aggregation phase. Haghani et al. extended Chan et al.'s protocol and enhanced its data availability (2008). The protocol allows the identification of nodes that caused the inconsistency in the aggregation result (or the aggregation disruption) and then allows the removal of malicious nodes. These nodes can be detected through successive polling of the layers on a commitment tree. Their protocol enhances security services provided by Chan et al.'s protocol (authentication, data integrity, and data freshness), and adds data availability.

Another protocol that considered data availability is proposed by Alzaid et al. (2008a). Their protocol integrated the aggregation functionalities with the advantages provided by a reputation system in order to enhance the network lifetime and the accuracy of the aggregated data without trimming the abnormal (but correct) readings. Eliminating abnormal readings with no further investigation is impractical, especially in applications such as monitoring bush fires or monitoring temperatures within oil refineries. The node behaviour is represented in the form of (α, β) tuple where α and β denote the amount of positive and negative ratings calculated by each node for other nodes in its cell (or cluster) and then stored in the reputation table. If node x has behaved well for a specific function, α_x is incremented by one. Otherwise, β_x is incremented. The nodes' behaviours are examined for three functions: data sensing, data forwarding, and data aggregation (if x is the cell representative for an intermediate cell). To fill the reputation table, each node evaluates the sensing, forwarding, and aggregation (if in an intermediate cell) functionalities and computes α and β for each function.

5. Security Analysis

This section provides the security analysis of current secure data aggregation protocols. This analysis can be difficult for the following reasons:

- Each protocol designers solved the data aggregation security from different angles. For example, some designers solved the problem by considering either single aggregator model or multiple aggregator model. Each model has its own challenges that need to be considered carefully. End-to-end encryption, for example, is easier to implement in the single aggregator model than the multiple aggregator model. However, the energy consumption at the single aggregator model has to be minimized in order to extend the network lifetime and enhance data availability service.

- There is no standard adversarial model where current secure data aggregation protocols compete to provide a higher level of security, or resilience to attacks discussed in Section 3.1. For example, secure data aggregation protocols that defeat type I adversary are secure in the face of SY, SF, and RE attacks. However, this resilience against these attacks is not provided by the protocol itself, but is due to the limited capabilities of type I adversary as discussed in Section 3.

Existing secure data aggregation protocols, consequently, are compared in a number of different ways: the aggregation model they follow, security services they provide, cryptographic primitives they use, and resilience against attacks described in section 3.1.

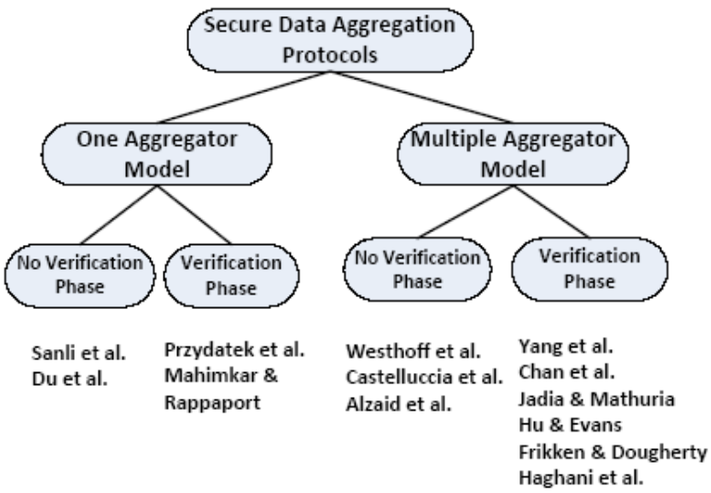


Fig. 4. Classification of current secure data aggregation protocols.

5.1 Aggregation Models

Based on our discussion in Section 4, current secure data aggregation protocols fall under either single aggregator model or multiple aggregator model. A sketch of these two aggregation models can be found in Figure 2. The aggregation process, in the single aggregator model, takes place once between the sensing nodes and the base station or the querier. All collected physical phenomena (PP) in WSNs, therefore, travel to only one aggregator point in the network before reaching the querier. On the other hand, collected data in WSNs are aggregated more than one time before reaching the final destination or the querier. This model achieves greater reduction in the number of bits transmitted within the network, especially in large WSNs. The importance of this model is growing as the network

size is getting bigger, especially when data redundancy at the lower levels is high. Figure 4 concludes the discussion in Section 4 and classifies secure aggregation protocols depending on the aggregation model they follow and whether they have a verification phase or not. This verification phase, if it exists, is used to validate the aggregation results (or the aggregator behaviour) by using some methods such as interactive protocols between the base station (or the querier) and normal sensor nodes.

5.2 Security Services

Since the considered adversarial model varies from one secure data aggregation protocol to another, as discussed in Section 3.3, each protocol provides different security services to defeat the expected type of adversary. Table 2 shows security services provided by each secure data aggregation protocol. It is obvious that protocols designed with type I adversary in mind, such as (Castelluccia et al., 2005; Sanli et al., 2004), do not provide authentication service while authentication is a must in protocols that defeat type II or type III adversaries as in (Alzaid et al., 2008a; Chan et al., 2006; Du et al., 2003; Frikken & IV, 2008; Haghani et al., 2008; Hu & Evans, 2003; Jadia & Mathuria, 2004; Mahimkar & Rappaport, 2004; Przydatek et al., 2003; Yang et al., 2006; Westhoff et al., 2006). As discussed in Section 3.3, type II and type III adversaries can launch, for example, SY attack where adversaries are able to present more than one node and then interact with the network. If authentication is not implemented, the adversaries can then successfully affect the overall aggregation results.

Scheme	CO	IN	FR	AV	AU	AT
Westhoff et al. (2006)	√				√	**II**
Hu & Evans (2003)		√	√		√	**II**
Przydatek et al. (2003)	√	√	√		√	**II**
Chan et al. (2006)		√	√		√	**III**
Du et al. (2003)		√			√	**II**
Mahimkar & Rappaport (2004)	√	√			√	**II**
Sani et al. (2004)	√		√			**I**
Yang et al. (2006)	√	√	√		√	**II**
Jadia & Mathuria (2004)	√	√	√		√	**II**
Castelluccia et al. (2005)	√					**I**
Frikken & Dougherty (2008)		√	√		√	**III**
Haghani et al. (2008)		√	√	√	√	**III**
Alzaid et al. (2008)		√	√	√	√	**II**

CO	Confidentiality	**IN**	Integrity
FR	Freshness	**AV**	Availability
AU	Authentication	**AT**	Adversary Type

Table 2. Security services provided in current secure data aggregation protocols.

Data confidentiality, furthermore, is provided in secure data aggregation protocols where the privacy of the data is important. Some of the protocols designers who considered type I adversary in their protocols (Castelluccia et al., 2005; Sanli et al., 2004) aimed to secure the raw data and the aggregated data from being revealed by the adversary. They focused on providing data confidentiality service only, and this level of security is acceptable where the adversary has no interest in destroying the overall performance but is interested in knowing the content of the reported information as in type I adversary. Other designers who considered type II or type III adversaries in their protocols (Jadia & Mathuria, 2004; Mahimkar & Rappaport, 2004; Przydatek et al., 2003; Yang et al., 2006; Westhoff et al., 2006) provide data confidentiality service in conjunction with other services to protect data privacy and strengthen their protocols' resilience against attacks that can be launched by the considered adversary model (type II or type III adversary).

Data integrity, moreover, is provided in secure data aggregation protocols where the protocols designers considered type II or type III adversaries. These two types, as discussed in Section 3.3, can launch NC attack and are consequently able to alter the content of the data received from downstream nodes, and needs to be forwarded to upper stream nodes. If data integrity service is not offered by the protocol, upper stream nodes therefore have no idea about this alteration. Table 2 shows that most secure data aggregation protocols that have type II or type III adversary in mind, such as (Alzaid et al., 2008a; Chan et al., 2006; Du et al., 2003; Frikken & IV, 2008; Haghani et al., 2008; Hu & Evans, 2003; Jadia & Mathuria, 2004; Mahimkar & Rappaport, 2004; Przydatek et al., 2003; Yang et al., 2006) provide data integrity service. However, Westhoff et al.'s protocol does not offer data integrity although it is built with type II adversary in mind. This is because the protocol designers limited their discussion to data confidentiality only.

Data freshness, furthermore, is considered by some of the protocols designers when they constructed their protocols (Chan et al., 2006; Hu & Evans, 2003; Jadia & Mathuria, 2004; Przydatek et al., 2003; Yang et al., 2006) in order to defeat type II or type III adversary. These types of adversary, as discussed in Section 3.3, can launch different types of attacks such as RE attack. The adversary, in RE attack, can affect the aggregation result by simply replaying old messages into the network if data freshness is not provided. For example, the designers of the witness-based secure data aggregation protocol (Du et al., 2003) did not provide data freshness service as discussed in Section 4.1.1. Although witnesses help the base station (or the querier) to validate the aggregation results, the aggregator - if compromised- can mislead the base station by replaying old messages with valid (but old) proofs from the witnesses.

Finally, data availability gained some attention from the protocols designers (Alzaid et al., 2008a; Haghani et al., 2008). Detecting the inconsistency in the aggregation results with no further action is not enough because the adversary can keep manipulating the aggregation result in order to bring the network down by consuming the energy resources of sensor nodes.

5.3 Cryptographic Primitives

The section lists cryptographic primitives used by the designers of secure data aggregation protocols to defeat the considered type of adversary. As discussed in Section 4, cryptographic primitives vary from one protocol to another depending on the type of adversary the protocols designers considered, and the security services they wanted their protocols to provide. Table 4 summarizes all security primitives used in the secure data aggregation protocols discussed in this chapter.

The message authentication code (*MAC*) is used to exclude unauthorized parties from sending forged aggregated data and to protect the original message from being altered in protocols (Chan et al., 2006; Du et al., 2003; Hu & Evans, 2003; Jadia & Mathuria, 2004; Przydatek et al., 2003; Yang et al., 2006). On the other hand, Mahimkar and Rappaport's protocol used digital signature (Mahimkar & Rappaport, 2004) and Castelluccia et al.'s (Castelluccia et al., 2005) and Westhoff et al.'s (Westhoff et al., 2006) protocols relied on privacy homomorphic encryption to prevent unauthorized parties from participating in the network, and affecting the data integrity of the aggregation result.

Scheme	MA	DS	SK	PK	RS	PH	BA	IP	VS	AT
Westhoff et al. (2006)			√			√				**II**
Hu & Evans (2003)	√		√				√			**II**
Przydatek et al. (2003)	√		√				√	√		**II**
Chan et al. (2006)	√		√				√	√		**III**
Du et al. (2003)	√		√						√	**II**
Mahimkar & Rappaport (2004)		√		√						**II**
Sani et al. (2004)			√							**I**
Yang et al. (2006)	√		√				√	√		**II**
Jadia & Mathuria (2004)	√		√							**II**
Castelluccia et al. (2005)			√			√				**I**
Frikken & Dougherty (2008)	√		√				√	√		**III**
Haghani et al. (2008)	√		√				√	√		**III**
Alzaid et al. (2008)	√				√					**II**

MA	Message Authentication	**DS**	Digital Signature
SK	Symmetric Key	**PK**	Public Key
RS	Reputation System	**PH**	Privacy Homomorphic
BA	Broadcast Authentication	**IP**	Interactive Protocol
VS	Voting Scheme	**AT**	Adversary Type

Table 4. Cryptographic primitives used in current secure data aggregation protocols

Symmetric and public key cryptography are used to achieve either hop-by-hop or end-to-end encryption whenever data confidentiality is required. Table 4 shows that all secure data aggregation protocols, discussed in this chapter, except Mahimkar and Rappaport's protocol. It used symmetric key cryptography. Mahimkar and Rappaport's protocol used elliptic curve cryptography (public key cryptography) to implement the encryption and the digital signature.

As discussed in Section 4, secure data aggregation protocols may or may not have a verification phase in order to check the validity of the aggregation results. The verification phase was designed using one of the following methods: an authenticated broadcast such as μTESLA (Hu & Evans, 2003), interactive proofs (Chan et al., 2006; Frikken & IV, 2008; Haghani et al., 2008; Yang et al., 2006; Przydatek et al., 2003), or voting systems (Du et al., 2003). The security primitives' subsections in Section 4 provide more details about these verification options.

5.4 Attack Visibility

This section concludes the attack visibility analysis that is discussed in the adversarial model and attack resistance subsections in Section 4. Secure data aggregation protocols, presented in this chapter, are investigated to determine whether or not they are vulnerable to different types of attack listed in Section 3.1.

Due to the communication nature in WSNs, only adversary of types II and III can launch DoS attack by sending radio signals that interfere with the radio frequencies used by WSNs. Another form of DoS attack occurs when the adversary refuses to compute (or forward) aggregation information and starts dropping messages when it succeeds in compromising a sensor node. Table 6 shows that all secure data aggregation protocols are vulnerable to DoS attack, especially its first form.

Scheme	DoS	NC	SY	SF	RE	AT
Westhoff et al. (2006)	√	√		√	√	**II**
Hu & Evans (2003)	√	√		√		**II**
Przydatek et al. (2003)	√	√		√		**II**
Chan et al. (2006)	√	√		√		**III**
Du et al. (2003)	√	√	√	√	√	**II**
Mahimkar & Rappaport (2004)	√	√		√	√	**II**
Sani et al. (2004)						**I**

Yang et al. (2006)	√	√	√	**II**
Jadia & Mathuria (2004)	√	√	√	**II**
Castelluccia et al. (2005)				**I**
Frikken & Dougherty (2008)	√	√	√	**III**
Haghani et al. (2008)	√	√		**III**
Alzaid et al. (2008)	√	√		**II**

DoS	Denial of Service	**NC**	Node Compromise
SF	Selective Forwarding	**SY**	Sybil
AC	Adversary Type	**RE**	Replay

Table 6. Attacks visibility in current secure data aggregation protocols.

Moreover, NC attack explains whether or not the adversary is able to reach any deployed sensor nodes and extracts its information stored in its memory. The NC attack is visible in all secure data aggregation protocols except for Sanli et al.'s and Castelluccia et al.'s protocols because these two protocols only considered type I adversary. In other words, NC attack is not visible in type I due to its limited capability as discussed in Section 3. It is worth mentioning that we classify the adversary considered in Westhoff et al.'s protocol into type II category although the designers aimed initially to defend passive adversary in their previous protocol (Girao et al., 2005). They then extended the adversary capability to launch NC attack against aggregator nodes.

As the capability of the adversary varies from type I to type III, the damage caused by these attacks also varies. Type I adversary, as discussed in Section 3.3, has not enough power to launch SY, SF, NC attacks. Therefore, SY and SF attacks are not visible in protocols (Castelluccia et al., 2005; Sanli et al., 2004) because of the adversary capability, not because of the security primitives the protocols designers used. SY attack is visible only in Du et al.'s protocol because leaf nodes are not authenticated to the aggregator (Du et al., 2003). An adversary, upon compromising a leaf node, can present more than one identity and then mislead the aggregator about the aggregation results, as discussed in Section 4.1.1.

Once the NC attack is visible in the network, this means the adversary has full control of the compromised node and can then selectively drop messages (SF attack). All secure data aggregation protocols, which considered type II and type III adversaries, are vulnerable to SF attack except for Haghani et al.'s and Alzaid et al.'s protocols. The former protocol has the adversary localizer component that marks nodes that disrupted the acknowledgment collection, and can then detect any SF attack activity (Haghani et al., 2008). The latter protocol computes nodes' reputation values for sensing, forwarding, and aggregating activities. Once the adversary has launched SF attack, the node's reputation value is reduced. If its reputation value falls below a threshold value due to performing malicious activities, the node is then black-listed (Alzaid et al., 2008).

Finally, RE attack occurs when the adversary has the ability to re-inject (or replay) old messages without even understanding its content. Most secure data aggregation protocols are resistant to this type of attack except (Castelluccia et al., 2005; Du et al., 2003; Mahimkar & Rappaport, 2004; Sanli et al., 2004; Westhoff et al., 2006). Surprisingly, RE attack is visible

in Du et al.'s, and Mahimkar and Rappaport's protocols (Du et al., 2003; Mahimkar & Rappaport, 2004) although they defeat type II adversary and the visibility of NC attack is considered. For example, once the adversary has compromised the aggregator node in Du et al.'s protocol, it is able to replay an old aggregation result with its valid proofs instead of the current result to mislead the base station. The adversary in Mahimkar and Rappaport's protocol can replay old valid signed aggregation results to mislead the base station when it succeeds in compromising the aggregator. The adversary in Westhoff et al.'s protocol can replay old encrypted messages once the compromise of an aggregator node is succeeded, which affects the aggregation results.

5.4 Framework for Evaluating New Schemes

Based on the analysis provided in the previous sections, a conceptual framework for secure data aggregation protocols is proposed. The framework helps the designers of new secure data aggregation protocols to strengthen their new design in the face of the adversary. To the best of our knowledge, this framework is the first work that establishes a common ground to compare different secure data aggregation protocols and draws the security map for new protocols.

Figure 5 suggests the minimum security requirements that a new protocol should maintain. The designers need to first study the adversary capability and then estimate the network size where the protocol will run.

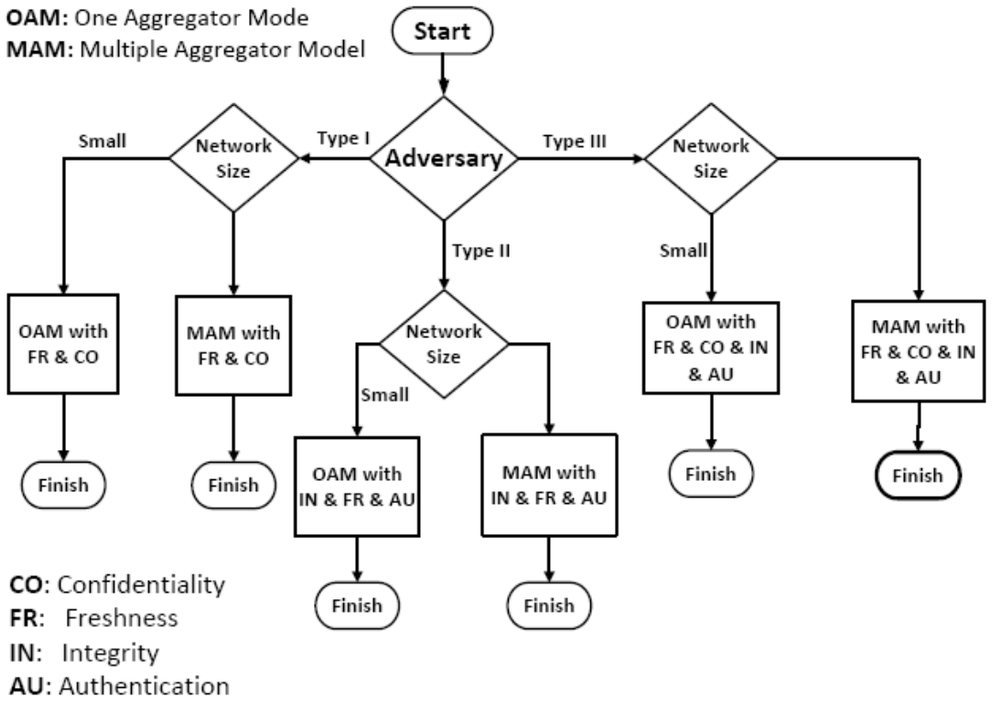


Fig. 5. The proposed framework.

Once the designers decided to defend type I adversary, they need to design a protocol that is at least resistant to passive adversary activities. As discussed in Section 3, type I adversary

maybe able to eavesdrop on traffic to obtain some knowledge about aggregated data. Thus, the protocol should at least provide data confidentiality. However, data confidentiality is application-dependent and is offered only when data privacy is needed. Data integrity, data freshness, and authenticity are not included with the minimum security requirement because type I adversary has not enough power to interact with the network and launch NC, SF, RE, DoS, and SY attacks in order to affect the overall performance of secure aggregation protocol.

Moreover, the designers of new protocols may consider type II or type III adversaries that have stronger capabilities than type I adversary. These adversaries can launch any type of attack listed in Section 3.1 in order to mislead the base station about the reported aggregation results. To defeat type II adversary, the framework in Figure 5 suggests that new secure data aggregation protocols should provide data integrity as well as data freshness, and authentication. As the adversary becomes stronger, the minimum security requirement should be enhanced by new services in order to provide resiliency against the adversary's attack. The framework suggests hiding the data (or providing data confidentiality) as well as authentication, data integrity, and data freshness.

The designers of new protocols should then consider the network size to decide whether to follow the one aggregator model or multiple aggregator model. The multiple aggregator model achieves greater reduction in the number of bits transmitted within the network especially in large WSNs, as illustrated in Figure 1. The importance of this model is growing as the network size is getting bigger, especially when data redundancy at the lower levels is high. In the following section, the performance analysis of selected secure data aggregation protocols is discussed.

6. Performance Analysis

This section provides the performance analysis of current secure data aggregation protocols in WSNs. Due to lack of space, we limit our discussion to the communication overhead only.

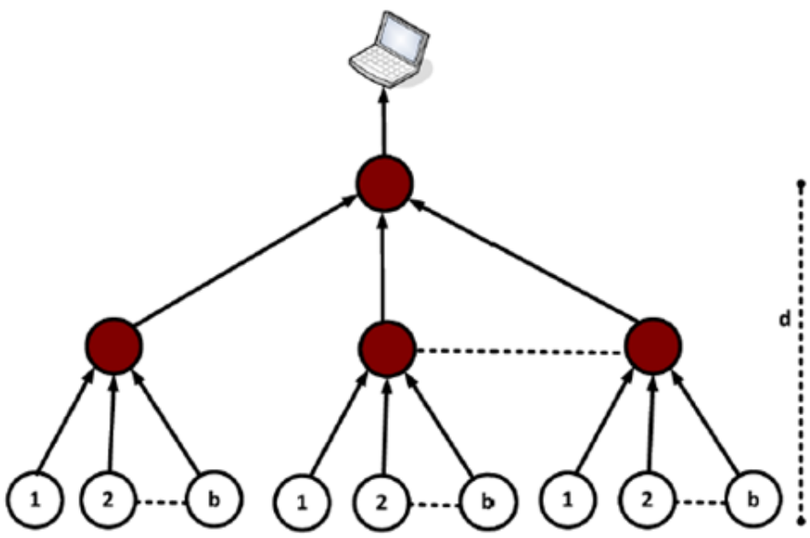


Fig. 6. The tree model used to analyze the performance of current secure data aggregation protocols.

This analysis focuses on calculating the number of bits transmitted within the network in order to show which secure data aggregation protocol is energy hungry and sends more information to accomplish the protocol objectives. We discuss seven scenarios where both aggregation models (single and multiple) are covered. These scenarios are: no aggregation, aggregation but no security, Hu and Evans's protocol (2003), Jadia and Mathuria's protocol (2004), Yang et al.'s protocol (2006), Przydatek et al.'s protocol (2003), and Du et al.'s protocol (2003). Since these scenarios may or may not have a verification phase, we limit our analysis to the aggregation phase only.

For concreteness, we consider an aggregation tree where its depth is d and each node (except leaf nodes) has b children as shown in Figure 6. This means the distance between the base station and leaf nodes are d, where d starts with zero at the first level. The total number of nodes (N) in this type of tree is n bits long and can be computed as:

$$N = \left(\frac{b^{d+1}+1}{b-1}\right) \tag{4}$$

This kind of tree, therefore, has b^d leaf nodes. If the scenario belongs to the single aggregator model, we consider the root of the tree to be the aggregator. Otherwise, any parent node acts as an aggregator (see Figure 6). In both models, each sensor node in the tree has to participate in the aggregation activity by sensing the environment and then reports its reading to upper nodes.

Let us explain some notations used in this section before we discuss those scenarios. Let x denote the length of the reported information (without the packet's header) where this information can be either raw data (reported from leaf nodes) or aggregated data (reported from the aggregator nodes).

Also, let y denote the length of the sensor node ID in bits, z denote the MAC's length in bits, and qn denote the length of the query nonce in bits. Moreover, TinyOS packet is preconfigured with a maximum size of 35 byte (29 byte payload and 6 byte header) and thus we denote the packet header by h.

6.1 First Scenario (No Aggregation & No Security)

We analyze the number of transmitted bits by considering the situation where no aggregation and no security are used within our example summarized in Figure 6. Leaf sensor nodes sense some physical phenomenon and report them to the upper nodes (their parents). The parents subsequently forward this information to upper nodes until the information is delivered and collected by the base station (or the querier). Each reported information contains the sensor node ID and the sensed physical phenomenon, which required each sensor node at level d to send $x + y + h$ bits long message to its parent. Each parent (intermediate node) needs to forward ($x + y + h$) bits for each child it has and ($x + y + h$) bits to report its reading. Thus, the total number of bits forwarded by each parent at level $d - i$ (where $i = d - 1$) is:

$$(b + 1)(x + y + h) \tag{5}$$

The total number of bits travelled in the network can be estimated from equation 5 as follows:

$$\sum_{i=0}^{d}(d - i)\, b^{(d-i)}\, (x + y + h) \tag{6}$$

6.2 Second Scenario (Aggregation but No Security)

We analyze and calculate the length in bits for transmitted information in the case where no security is provided in our example but the aggregation functionality is implemented. This scenario is similar to the example discussed in Section 2. Each parent, in this scenario, combines the reported b messages from its children and reports only one message that represents these b messages. The number of bits forwarded by each parent at any level is estimated as $x + y + h$ and the total number of bits, travelled in the network in order to accomplish the aggregation phase, is calculated as:

$$\left(\frac{b^{(d+1)}-1}{b-1}\right)(x + y + h) \tag{7}$$

6.3 Third Scenario (Hu & Evans)

We analyze the protocol by Hu and Evans (2003). The protocol, as discussed in Section 4, follows the multiple aggregator model with a verification phase. Each leaf node (at level $d - i$ where $i = 0$) needs to send its ID, data, and one message authentication code toward its parent. The length of this message in bits is $x + y + z + h$. The total number of bits that the protocol requires leaf nodes to send to their parents (at level $d - i$ where $i = 0$) is:

$$b^d(x + y + z + h) \tag{8}$$

Each parent (at levels $d - i$ where $i = 1, 2, ..., d$) needs to forward the received data unchanged and add one more *MAC*. Thus, the length of this message in bits can be calculated as $b(x + y + z) + z + h$ and the total number of bits sent by all parents is:

$$\sum_{i=1}^{d} b^{(d-i)}[b(x + y + z) + z + h] \tag{9}$$

Thus, the approximate number of bits transmitted to perform the aggregation phase, in this Protocol, can be calculated by adding equation 8 and equation 9 together as follows:

$$b^d(x + y + z + h) + \sum_{i=1}^{d} b^{(d-i)}[b(x + y + z) + z + h]$$
$$= b^d(x + y + z + h) + \left(\frac{b^{(d+1)}-1}{b-1} - b^d\right)[b(x + y + z) + z + h] \tag{10}$$

6.4 Fourth Scenario (Jadia & Mathuria)

The improvement done by Jadia and Mathuria's protocol (2004), in order to add data confidentiality service to the security services provided by Hu and Evans's protocol, requires each node to add one more message authentication into each message. So, each sensor node (at level $d - i$ where $i = 0$) sends $x + y + 2z + h$ bits instead of sending $x + y + z + h$ bits in Hu and Evans's protocol (Hu & Evans, 2003). Therefore, the total number of bits sent by all leaf nodes is $b^d(x + y + 2z + h)$ and the total number of bits sent by the protocol to accomplish the aggregation function is approximately:

$$b^d(x+y+2z+h)+\sum_{i=1}^{d} b^{(d-i)}[b(x+y+z)+z+h]$$

$$= b^d(x+y+2z+h)+\left(\frac{b^{(d+1)}-1}{b-1}-b^d\right)[b(x+y+z)+z+h] \quad (11)$$

6.5 Fifth Scenario (Yang et al.)

Yang et al., as discussed in Section 4, followed the multiple aggregator model and used the divide-and-conquer principle to divide the network tree into multiple logical subtrees, which increases the number of aggregators and reduces the number of nodes in each subtree. For simplicity, we assume that the total number of sensor nodes is *N* and each subtree has an average size of *s* sensors. The number of subtrees, therefore, is $\frac{N}{s}+1$ considering the base station as a subtree. Also, the height of a subtree can be approximated by $\frac{d}{2}$ and the distance from each subtree's leader to the base station is $\frac{d}{2}$. Each leaf node needs to send its ID, aggregation flag (one bit), an encrypted sensed data concatenated with a *MAC*, and the query sequence number. This transmission is about $x + y + z + 1 + h$ bits long. Therefore, the total number of bits transmitted in each subtree (or group) can be calculated as:

$$(s-1)(x+y+z+1+h)$$

The distance between the subtree's leader and the base station varies, depending on the position of the subtree. It can be anything between $[0, \frac{d}{2}]$ and for simplicity we assume that the distance between all subtrees' leaders and the base station is $\frac{d}{4}$. Each subtree's leader forwards the aggregation result toward the base station and this increases the number of travelled bits within the network by $\left(\frac{N}{s}\right)\left(\frac{d}{4}\right)(x+y+z+1+h)$ bits. Therefore, the total number of bits sent across the network to accomplish the aggregation function is approximated by:

$$\left(\frac{N}{s}\right)(s-1)(x+y+z+1+h)+\left(\frac{N}{s}\right)\left(\frac{d}{4}\right)(x+y+z+1+h)$$
$$= \left(\frac{N}{s}\right)(x+y+z+1+h)\left[(s-1)+\frac{d}{4}\right] \quad (12)$$

6.6 Sixth Scenario (Przydatek et al.)

We analyze the number of transmitted bits across the network in order to accomplish the aggregation function in Przydatek et al.'s protocol (Przydatek et al., 2003). Their protocol used the aggregate-commit-prove approach discussed in Section 4.1.2. In the aggregate phase, each sensor needs to send its ID, data, query nonce, and two message authentication codes keyed with two shared keys: the first key is shared with the aggregator and the other key is shared with the base station. The length of this message in bits is $x + y + qn + 2z + h$ and it travels all the way toward the aggregator. Therefore, the total number of bits travelled within the network until the sensed data reaches the aggregator is:

$$\sum_{i=0}^{d}(d-i)\ b^{(d-i)}(x+y+qn+2z+h) \tag{13}$$

In the commit phase, the aggregator constructs a Merkle hash tree of the received messages and sends the root of this tree as a commitment value, the number of leaves in the hash tree, and aggregated result. Let us assume for simplicity the length of the commitment value is $x+y+qn+2z+h$ bits long and the length of the aggregated result as long as the reported data x. Thus, the total number of bits sent to the home server (or remote user) by the aggregator is:

$$n+2x+y+qn+2z+h \tag{14}$$

Adding the number of bits in equations 13 and 14 gives the total number of travelled bits required to perform the aggregation function in this protocol as follows:

$$n+2x+y+qn+2z+h+\sum_{i=0}^{d}(d-i)\ b^{(d-i)}(x+y+qn+2z+h) \tag{15}$$

6.7 Seventh Scenario (Du et al.)

In Du et al.'s protocol, the designers assumed that leaf nodes are honest and the sensed data reaches the aggregator and witnesses correctly. Let us assume that each sensor needs to send at least its ID and its sensed data. The length of this message in bits is $x + y + h$. Therefore, the total number of bits travelling within the network to reach the aggregator for each event is:

$$\sum_{i=0}^{d}(d-i)\ b^{(d-i)}(x+y+h) \tag{16}$$

Also, the same number of bits goes to each witness (w) and consequently the total number of travelled bits is:

$$w\sum_{i=0}^{d}(d-i)\ b^{(d-i)}(x+y+h) \tag{17}$$

Each witness computes the aggregation result and sends to the aggregator the message authentication code (*MAC*) that contains its ID and aggregation result. Finally, the aggregator forwards its ID, aggregation result (computed by itself), and all *MAC*s received from the witnesses. Therefore, the total number of travelled bits is:

$$\begin{aligned}\sum_{i=0}^{d}(d-i)\ b^{(d-i)}(x+y+h)\ +\ w\sum_{i=0}^{d}(d-i)\ b^{(d-i)}(x+y+h)+\\ w(z+h)+(x+y+wz+h)\\ =\ 2wz+x+y+h(w+1)+\ (w+1)\sum_{i=0}^{d}(d-i)b^{(d-i)}(x+y+h)\end{aligned} \tag{18}$$

6.8 Example

For better understanding the transmission overhead caused by secure data aggregation protocols chosen in the above scenarios, we give an example with numbers. Let us select the length of the reported information without the header (x), the length of the sensor ID in bits (y), the MAC's length in bits (z), the number of witnesses (w), the length in bits for the average number of sensors in any subtree (s), the length of the query number in bits (qn), and the length in bits for the total number of sensor nodes (n) to be 7 bytes, 2 bytes, 6 bytes,

5 witnesses, 1 byte, 3 bytes, and 4 bytes respectively. We compute the number of bytes that each secure aggregation protocol transmits to accomplish the aggregation phase by substituting the values given above into equations 6, 7, 10, 11, 12, 15, and 18. Table 7 investigates our scenarios and substitutes variables with numbers to give a clearer idea.

Scenarios	*b=2*		*b=3*		*b=4*	
	d=3	*d=4*	*d=3*	*d=4*	*d=3*	*d=4*
First Scenario (No Aggregation & No Security)	510	1470	1530	6390	3420	18780
Second Scenario (Aggregation but No Security)	225	465	600	1815	1275	5115
Third Scenario (Hu & Evans, 2003)	462	966	1113	3381	2226	8946
Fourth Scenario (Jadia & Mathuria, 2004)	510	1062	1275	3867	2610	10482
Fifth Scenario (Yang et al., 2006)	317	682	844	2662	1792	7502
Sixth Scenario (Przydatek et al., 2003)	1061	2981	3101	12821	6881	37601
Seventh Scenario (Du et al., 2003)	3165	8925	9285	38445	20625	112785

Table 7. Number of bytes transmitted across the network to accomplish the aggregation phase.

7. Conclusion

This chapter gives a detailed review of secure data aggregation protocols in wireless sensor networks. It first explains the motivation behind secure data aggregation and discusses the security requirements of wireless sensor networks required to strengthen attack-resistant data aggregation protocols. It then describes the adversarial model that can threaten any secure aggregation protocol. The different capabilities an adversary may have against secure data aggregation protocols are discussed. After that, the state-of-the-art in secure data aggregation protocols is surveyed and classified into two categories (one aggregator model and multiple aggregator model) based on the number of aggregator nodes and the existence of the verification phase. To provide the security and performance analysis, current secure data aggregation protocols are compared in a number of different ways: the aggregation model they follow, security services they provide, cryptographic primitives they use, attacks they secure against, and the number of bits they require nodes to send in order to accomplish the aggregation phase. Based on this security and performance analysis, a conceptual framework that leads to better evaluation of secure aggregation schemes is given.

8. References

Alzaid, H., Foo, E. & Nieto, J. G. (2008a). RSDA: Reputation-based secure data aggregation in wireless sensor networks, *Proceedings of the 9th International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT'08*, pp. 419–424, Dunedin, New Zealand, December, 2008, IEEE Computer Society.

Alzaid, H., Foo, E. & Nieto, J. G. (2008b). Secure data aggregation in wireless sensor network: a survey, *Proceedings of the 6th Australasian conference on Information security, AISC'08*, pp. 93–105, Wollongong, NSW, Australia, January, 2008, Australian Computer Society.

Castelluccia, C., Mykletun, E. & Tsudik, G. (2005). Efficient aggregation of encrypted data in wireless sensor networks, *Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems, MobiQuitous'05*, pp. 109-117, San Diego, CA, USA, July, 2005, IEEE Computer Society.

Chan, H., Perrig, A. & Song, D. (2006). Secure hierarchical in network aggregation in sensor networks, *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS'06*, pp. 278–287, Alexandria, VA, USA, November, 2006, ACM.

Crossbow Technology Inc. (2006). Mica2 datasheet. *Crossbow Technology Inc*. Retrieved October 13, 2009, from: http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.

Domingo-Ferrer, J. (2002). A provably secure additive and multiplicative privacy homomorphism, *Proceedings of the 5th International Conference on Information Security, ISC'02*, Vol. 2433 of Lecture Notes in Computer Science, pp. 471–483, Sao Paulo, Brazil, October, 2002, Springer .

Du, W., Deng, J., Han, Y. S. & Varshney, P. (2003). A witness-based approach for data fusion assurance in wireless sensor networks, *Proceedings of IEEE Global Communications Conference, GLOBECOM'03*, Vol. 3, pp. 1435–1439, San Francisco, USA, December, 2003, IEEE Computer Society.

Frikken, K. B. & IV, J. A. D. (2008). An efficient integrity-preserving scheme for hierarchical sensor aggregation, *Proceedings of the First ACM Conference on Wireless Network Security, WISEC'08*, pp. 68–76, Alexandria, VA, USA, April, 2008, ACM.

Girao, J. , Westhoff, D., Schneider, S. (2005). CDA: Concealed Data Aggregation for Reverse Multicast Traffic in Wireless Sensor Networks, *Proceedings of IEEE International Conference on Communications, ICC'05*, Vol. 5, pp. 3044–3049, Seoul, Korea, May, 2005, IEEE.

Grigoriev, D. & Ponomarenko, I. V. (2003). Homomorphic public key cryptosystems over groups and rings, *The Computing Research Repository, CoRR,* Vol. cs.CR/0309010, September, 2003, Cornell University.

Grubbs, F. (1969). Procedures for detecting outlying observations in samples, Technometrics Vol. 11, No. 1, pp. 1–21, February, 1969, American Statistical Association.

Guimarães, G., Souto, E., Sadok, D. F. H. & Kelner, J. (2005). Evaluation of security mechanisms in wireless sensor networks, *Proceedings of the 2005 Systems Communications (ICW/ICHSN/ICMCS/SENET)*, pp. 428–433, Montreal, Canada, August, 2005, IEEE Computer Society.

Haghani, P., Papadimitratos, P., Poturalski, M., Aberer, K. & Hubaux, J.-P. (2008). Efficient and robust secure aggregation for sensor networks, *The Computing Research Repository (CoRR)*, Vol. CoRR abs/0808.2676, August, 2008, Cornell University.

He, T., Vicaire, P., Yan, T., Luo, L., Gu, L., Zhou, G., Stoleru, R., Cao, Q., Stankovic, J. A. & Abdelzaher, T. F. (2006). Achieving real-time target tracking using wireless sensor networks, *Symposium of the 12th IEEE Real-Time and Embedded Technology and Applications, RTAS'06*, pp. 37–48, San Jose, California, USA, April, 2006, IEEE Computer Society.

Hu, L. & Evans, D. (2003). Secure aggregation for wireless network, *Symposium on Applications and the Internet Workshops, SAINT'03*, pp. 384–394, Orlando, FL, USA, January, 2003, IEEE Computer Society.

Jadia, P. & Mathuria, A. (2004). Efficient secure aggregation in sensor networks, *Proceedings of the 11th conference on High Performance Computing, HiPC'04*, Vol. 3296 of Lecture Notes in Computer Science, pp. 40-49, Bangalore, India. December, 2004, Springer.

Krishnamachari, B., Estrin, D. & Wicker, S. B. (2002). The impact of data aggregation in wireless sensor networks, *Proceedings of the 22nd International Conference on Distributed Computing Systems, Workshops, ICDCSW'02*, pp. 575–578, Vienna, Austria, July, 2002, IEEE Computer Society.

Law, Y. W., Doumen, J. & Hartel, P. H. (2006). Survey and benchmark of block ciphers for wireless sensor networks, *ACM Transaction on Sensor Networks, TOSN*, Vol. 2, No. 1, pp. 65–93, February, 2006, ACM.

Mahimkar, A. & Rappaport, T. S. (2004). SecureDAV: A secure data aggregation and verification protocol for sensor networks, *Proceedings of IEEE Global Communications Conference, GLOBECOM'04*, Vol. 4, pp. 2175– 2179, Dallas, Texas, USA, December, 2004, IEEE Computer Society.

Mainwaring, A. M., Culler, D. E., Polastre, J., Szewczyk, R. & Anderson, J. (2002). Wireless sensor networks for habitat monitoring, *Proceedings of the First ACM International Workshop on Wireless Sensor Networks and Applications, WSNA'02*, pp. 88–97, Atlanta, Georgia, USA, September, 2002, ACM.

Merkle, R. C. (1980). Protocols for public key cryptosystems, *IEEE Symposium on Security and Privacy*, pp. 122–134, Atlanta, CA, USA, April, 1980, IEEE Computer Society.

Murthy, C. S. R. & Manoj, B. (2004). *Ad Hoc Wireless Sensor Networks Architectures and Protocols*, Prentice Hall PTR, ISBN 978-0-13-147023-1, Upper Saddle River, NJ, USA.

Ozdemir, S. & Xiao, Y. (2009). Secure data aggregation in wireless sensor networks: A comprehensive overview, *Computer Networks*, Vol. 53, No. 12, pp. 2022-2037, August, 2009, Elsevier.

Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V. & Culler, D. E. (2002). SPINS: security protocols for sensor networks, *Wireless Network*, Vol. 8, No. 5, pp. 521–534, September, 2002, Springer.

Przydatek, B., Song, D. X. & Perrig, A. (2003). SIA: secure information aggregation in sensor networks, *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys '03*, pp. 255–265, Los Angeles, California, USA, November, 2003, ACM.

Rivest, R. L., Shamir, A. & Adleman, L. M. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *Communication of the ACM*. Vol. 21, No. 2, pp. 120–126, February, 1978, ACM.

Roosta, T., Shieh, S. & Sastry, S. (2006). Taxonomy of security attacks in sensor networks, *The First IEEE International Conference on System Integration and Reliability Improvements, SIRI'06*, pp. 13-22, Hanoi, Vietnam, December, 2006, IEEE Computer Society.

Sang, Y., Shen, H., Inoguchi, Y., Tan, Y. & Xiong, N. (2006). Secure data aggregation in wireless sensor networks: a survey, *Proceedings of the Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, PDCAT'06*, pp. 315–320, Taipei, Taiwan, December, 2006, IEEE Computer Society.

Sanli, H. O., Ozdemir, S. & Cam, H. (2004). SRDA: secure reference-based data aggregation protocol for wireless sensor networks, *Proceeding of the IEEE 60th Vehicular Technology Conference, VTC'04*, pp. 4650– 4654, Los Angeles, USA, September, 2004, IEEE Computer Society.

Setia, S., Roy, S. & Jajodia, S. (2008). Secure data aggregation in wireless sensor networks, In *Wireless Sensor Network Security*, in J. Lopez & J. Zhou (eds), chapter 8, pp. 204–222, April, 2008, IOS Press, ISBN 978-1586038137, Amsterdam, The Netherlands.

Shi, E. & Perrig, A. (2004). Designing secure sensor networks, *IEEE Wireless Communications*, Vol. 11, No. 6, pp. 38–43, December, 2004, IEEE Computer Society.

Wagner, D. (2003). Cryptanalysis of an algebraic privacy homomorphism., in C. Boyd & W. Mao (eds), *Proceedings of the Sixth International Conference on Information Security, ISC'03*, Vol. 2851 of Lecture Notes in Computer Science, pp. 234– 239, Bristol, UK, October, 2003, Springer.

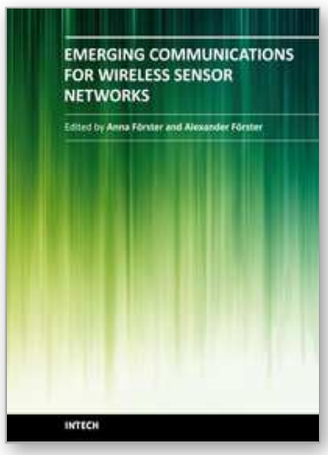
Wagner, D. (2004). Resilient aggregation in sensor networks, *Proceedings of the 2nd ACM Workshop on Security of ad hoc and Sensor Networks, SASN '04*, pp. 78–87, Washington DC, USA, October, 2004, ACM.

Westhoff, D., Girao, J. & Acharya, M. (2006). Concealed data aggregation for reverse multicast traffic in sensor networks: encryption, key distribution, and routing adaptation, *IEEE Transactions on Mobile Computing*, Vol. 5, No. 10, pp. 1417–1431, October, 2006, IEEE.

Yang, Y., Wang, X., Zhu, S. & Cao, G. (2006). SDAP: a secure hop-by-hop data aggregation protocol for sensor networks, *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc'06*, pp. 356–367, Florence, Italy, May, 2006, ACM.

Yick, J., Mukherjee, B. & Ghosal, D. (2008). Wireless sensor network survey, *Computer Networks*, Vol. 52, No. 12, pp. 2292–2330, August, 2008, Elsevier.

As a result, those bottleneck nodes around the sink deplete their batteries much faster than other nodes and, therefore, their lifetime upper bounds the lifetime of the whole network.



Emerging Communications for Wireless Sensor Networks

Edited by

ISBN 978-953-307-082-7
Hard cover, 270 pages
Publisher InTech
Published online 07, February, 2011
Published in print edition February, 2011

Wireless sensor networks are deployed in a rapidly increasing number of arenas, with uses ranging from healthcare monitoring to industrial and environmental safety, as well as new ubiquitous computing devices that are becoming ever more pervasive in our interconnected society. This book presents a range of exciting developments in software communication technologies including some novel applications, such as in high altitude systems, ground heat exchangers and body sensor networks. Authors from leading institutions on four continents present their latest findings in the spirit of exchanging information and stimulating discussion in the WSN community worldwide.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hani Alzaid, Ernest Foo, Juan Gonzalez Neito and DongGook Park (2011). Security Data Aggregation in Wireless Sensor Networks, Emerging Communications for Wireless Sensor Networks, (Ed.), ISBN: 978-953-307-082-7, InTech, Available from: http://www.intechopen.com/books/emerging-communications-for-wireless-sensor-networks/security-data-aggregation-in-wireless-sensor-networks

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen