Journal of Universal Computer Science, vol. 17, no. 11 (2011), 1550-1571
submitted: 30/9/10, accepted: 21/6/11, appeared: 1/7/11 © *J.UCS*

RAHIM: Robust Adaptive Approach Based on Hierarchical Monitoring Providing Trust Aggregation for Wireless Sensor Networks

Nabila Labraoui
(STIC University of Tlemcen, Tlemcen, Algeria
labraouinabila@yahoo.fr)

Mourad Gueroui
(PRISM University of Versailles, Versailles, France
mourad.gueroui@prism.uvsq.fr)

Makhlouf Aliouat
(University of Setif, Setif, Algeria
m_aliouat@yahoo.fr)

Jonathan Petit
(University of Twente, Enschede, The Netherlands
j.petit@utwente.nl)

Abstract: In-network data aggregation has a great impact on the energy consumption in large-scale wireless sensor networks. However, the resource constraints and vulnerable deployment environments challenge the application of this technique in terms of security and efficiency. A compromised node may forge arbitrary aggregation value and mislead the base station into trusting a false reading. In this paper, we present RAHIM, a *reactive defense* to secure data aggregation scheme in cluster-based wireless sensor networks. The proposed scheme is based on a novel application of adaptive hierarchical level of monitoring providing accuracy of data aggregation result in lightweight manner, even if all aggregator nodes and a part of sensors are compromised in the network.

Keywords: Accuracy, Availability, Data aggregation, Monitoring mechanism, Wireless sensor networks, Security
Categories: C.2, C.2.3

1 Introduction

Wireless sensor networks (WSN) are rapidly emerging technologies with potentials for many different distributed applications, such as detection of chemical or biological agents, fire detection or tracking of enemy vehicles, which renders them a hot research topic over the past few years. However, sensor network has extremely constrained resources like energy, bandwidth and capabilities of processing and storing data. The current version of sensors such as mica2 [Corporation, 07] uses a 16 bits, 8 MHz Texas Instruments MSP430 microcontroller with only 10 KB RAM, 48 KB Program space, 1024 KB External flash, and is powered by two AA batteries.

Therefore, the key challenge in sensor networks is to maximize the lifetime of sensor nodes due to the fact that it is not easy to replace the batteries of thousands of sensor nodes quickly. Data aggregation technique can greatly help to conserve the scarce energy resources by eliminating data redundancy and minimizing the number of data transmission [Labraoui, 11]. However, the resource constraints and vulnerable deployment environments challenge the application of this technique in terms of security and efficiency.

Sensor networks are typically deployed in unsecured areas which make them vulnerable against physical node capture attacks in which intruders take control of one or more sensor nodes to subvert network's performance [Akyildiz, 02]. Capture of one sensor node reveals all the security and network information to the adversary. Then, the adversary can easily launch internal attacks with data alteration, message negligence, selective forwarding, jamming, etc [Maarouf, 09], [Ning, 05]. Considering the data aggregation scenario, the compromised nodes can successfully authenticate forged reports to their neighbors, which have no way to distinguish bogus data from legitimate ones [Perrig, 04]. It can also alter the aggregation result in order to fabricate a false event report to mislead the decision makers, or keep injecting bogus data to cause network outage. In critical applications, using incorrect or maliciously corrupted data can have disastrous consequences.

In our work, we focus on data integrity, which prevents the compromised source nodes or aggregator nodes from significantly altering the final aggregation value. However, the main drawbacks of existing solutions that focus on integrity of data aggregation are the *expensive cost* and *the total data rejection*. The expensive cost problem is due to the generation of some heavy communication, computation overhead, or to the requirement of expensive interactive verification between the base station (*BS*) and sensors. The second important problem is total data rejection. The violation of data integrity anywhere in the network obligates the *BS* to reject the received aggregation result leading to the cancellation of all steps in the aggregation process. Thus, an important amount of correct data is lost resulting in wasting precious network resource.

In this paper, we present a new framework called RAHIM (Robust Adaptive approach based on HIerarchical Monitoring) for solving the above problems and improving reliability and high availability of cluster-based WSN. The cornerstone of our proposal is the management of a new type of monitoring mechanism called hierarchical monitoring. This new type of monitoring, allows verifying the integrity and the accuracy of aggregation results in two levels only if necessary, i.e. only when cheating is detected. This allows the *BS* to receive the correct result even in presence of compromised nodes. Contrary to previous solutions, which have a unique management rule, our proposal has several management rules and adapts its reaction in function of attack scenario. The accuracy of aggregation and energy efficiency are the main design goal of our scheme.

To assess the practicality of the proposed framework, we present very encouraging results, which clearly demonstrate appreciable energy conservation and small overhead stemming from both monitoring and aggregation operations.

The rest of the paper is organized as follows: we begin in section 2 with related work. Network assumptions and threat model are presented in section 3. In section 4, the design goals of RAHIM are presented and in section 5, we describe our secure aggregation scheme. Section 6 provides a security analysis and section 7 provides a performance evaluation. We conclude our work in section 8.

2 Related work

Wireless sensor networks are operated in an open, publicly accessible, and untrusted environment. Therefore, integrity of data aggregation is a big concern. Several research initiatives exist in literature to address the security concerns in data aggregation, however reducing the security overheads and aggregation cost remains an open issue.

Hu and Evans [Hu, 03] proposed the first work that addresses the security problem in aggregation protocol (*SDA*) for WSNs that is resilient to both intruder devices and single device key compromises. They present a secure aggregation protocol to detect misbehaving sensor nodes by exploiting two main ideas: *delayed aggregation* and *delayed authentication*. Instead of performing aggregation at parent nodes, it is delayed one level above. This increases bandwidth but allows detecting single corrupted nodes. However, the protocol may be vulnerable if a parent and a child node in the hierarchy are compromised.

Przydatek et al. [Przydatek, 03] proposed *SIA* protocol. *SIA* addresses data integrity by constructing efficient random sampling mechanisms and interactive proofs to verify that the answer given by the aggregator (or cluster-head) is a good approximation of the true value. *SIA* is the first work on secure data aggregation in sensor networks that can handle malicious aggregators and sensor nodes. The drawback of this protocol is that the statistical security property is achieved under the assumption of a single-aggregator model, where sensor nodes send their data to a single-aggregator node. In this way, the interactive verification (or authentication) procedure results in additional bandwidth consumption.

Du et al. [Du, 03] proposed a witness-based data aggregation scheme (*WDA*) for WSNs to assure the validation of the data sent from aggregator nodes to the base station. In order to prove the validity of the aggregation result, the aggregator node has to provide proofs from several witnesses. A witness node gets the same input as the aggregator node and performs data aggregation, however, without forwarding the result. Instead, the witness computes the message authentication code (MAC) of the result and then provides it to the aggregator node that must forward the proofs to the *BS*. However this scheme incurs a very high overhead transmission even when there is no attack.

Yang et al. [Yang, 06] propose *SDAP* scheme based on a commit and attest paradigm. In the commit phase, nodes are divided in groups and each group provides the sink with the group aggregate, while nodes commit to their measurements. The sink uses the maximum normalized residual test to decide which groups provided suspicious results. During the attest phase, subsets of those nodes are required to provide their measurements. Because of the outlier detection technique, the protocol is suitable only to sensor networks where all groups sense similar values. Moreover, the commit and attest paradigm requires multiple messages to detect the presence of an attacker. Similar to *SIA*, the overhead for grouping, commitment and attestation can be large. In another interesting work [Cristofaro, 09], the authors propose Fuzzy-based framework (*FAIR*) for resilient data aggregation in real-time responsive wireless sensor networks supporting in-network processing. Like to Du's protocol [Du, 03], witness nodes have been often employed to confirm the result of aggregator nodes, in order to ensure the integrity of data during aggregation, and aggregate and forward the result themselves. However, witnesses not only confirm the aggregator's result, but also aggregate and forward the result themselves. Thus, the aggregator nodes on a higher level receive the full data and extract information even if the nodes disagree. Based on this data, the *BS* can apply fuzzy logic to decide about the correctness of the query result. This latter approach also addresses the possibility of malicious aggregator nodes manipulating data. However, this work induces overhead with the application of witness nodes.

Despite of the diversity and the proved efficiency of these solutions, they result in data rejection if data integrity is violated anywhere in the network. However, as long as such subversive activity exists, no aggregation result can be obtained. Thus, our work is motivated by investigating this crucial problem which is causing waste of precious network resources.

3 Network Assumptions and Threat Model

We consider a cluster-based sensor network that consists of *n* stationary sensor nodes and stationary base station (*BS*). Each sensor node has a unique identifier Id_i, $1 \le i \le n$. The network is divided into clusters, each of which has a cluster-head (*CH*). According to Sun et al. [Sun, 06] cluster formation protocol, inside each cluster (clique), each node is in the communication range of the remaining nodes of the cluster. Consequently, nodes of the same cluster can directly communicate, using one-hop communication only. Hence, while one sensor node is sending a message to *CH*, the message can be heard and received simultaneously by all other sensor nodes in the cluster, like watchdog in [Maarouf, 09]. We suppose that all nodes can directly reach the *BS* as supposed in LEACH protocol [Handy, 02]. In addition, and in order to minimize the communication overhead on the network, only *CHs* communicate directly with the *BS*, the remaining nodes communicate only with the nodes of their corresponding cluster. Nodes use two levels of communication power, a minimum power P_{min} when communicating between them inside the same cluster, and a higher power P_{max} when a *CH* communicating with the *BS*.

We assume that sensor nodes are similar to the current generation of sensor nodes, e.g., Mica2 motes, in their computational and communication capabilities and power resources, while the sink is a laptop class device supplied with long-lasting power.

We assume that there exists a reliable communication channel that sensor nodes can use to alert the *BS* of the presence of cheating, and its latency bound is known, i.e. we consider the availability of a method for sensor nodes to (reliably) communicate with *BS* without using the aggregator. This alarm channel is more expensive even than the link between the aggregator and the *BS*; however, since it is not used unless a cheating is detected, its high cost is not a factor under normal operation.

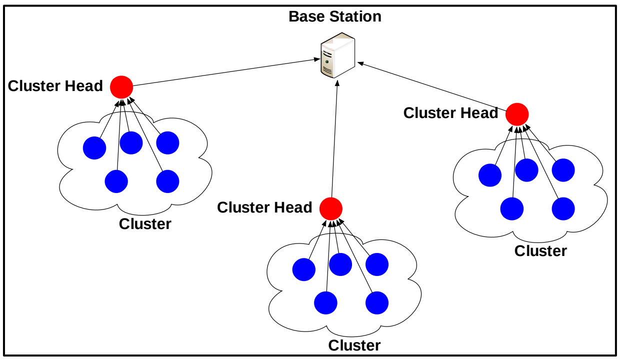

Figure 1: Network model

We assume that the attacker has control over an arbitrary number of sensor nodes, including knowledge of all their secret keys. The sole goal of the attacker is to launch a stealthy attack [Pai, 10], i.e. to cause the *BS* to accept a false aggregate that is higher or lower than the true aggregate value. This attack can be done either by direct injection attack or by false aggregation attack. We assume that an attacker can compromise at most t out of n nodes within the cluster ($t<n/2$). We assume that *BS* is trusted and cannot be compromised.

Table 1 summarizes the notation used in this work.

Notation	Description
BS	Base Station
CH	Cluster-Head which acts as an aggregator
PSUP_L1	Principal Monitor in first level
PSUP_L2	Principal Monitor in second level
$MONIT_i$	Second Level Monitor *i*
Id_i	Identifier of the sensor *i*
K_i^{BS}	Symmetric Key shared between sensor *i* and *BS*
$MAC_{K_i^j}(m)$	Message Authentication Code of message *m* with the key shared between *i* and *j*
AGG_i	Aggregation result calculated by sensor *i*
N_a	A nonce disseminated by *BS* when starting query

Table 1: Notation

4 Design Goals

Under the aforementioned conditions, a security concept is required to reduce the overhead of the aggregation alteration due to node compromission. Therefore, the proposed scheme has been designed with the following goals:

Accuracy: the aggregate result will be resilient against compromised nodes and data manipulation. Hence the result accepted by the base station will never deviate too far from the true value.

Availability: as long as the attack persists, the *BS* can obtain correct aggregate value even when all aggregators and some of sensors are compromised in the cluster.

Efficiency: the scheme will ensure the security goals in a lightweight manner. It generates low communication overhead and low energy consumption.

5 The Proposed secured scheme: RAHIM

In this section, we present our secure data aggregation scheme. We first give an overview of the protocol and then detail it.

5.1 Overview of the proposed scheme

The design of RAHIM is based on the principles of *independent aggregation* and *adaptive hierarchical level monitoring-based accuracy*. Our scheme is built on one core concept: no trust is supposed in any sensor. Therefore we design two hierarchical levels monitoring to ensure the integrity and the accuracy of the aggregate result. In the first level monitoring, we dedicate a sensor node to act as a principal supervisor (*PSUP_L1*). This *PSUP_L1* monitors the behavior of cluster-head (*CH*). Whereas in the second level monitoring, the rest of sensor nodes in the cluster act as peer monitors and monitor the behavior of both of *PSUP_L1* and *CH*. For efficiency, we dedicate among these peer monitors, a principal supervisor (*PSUP_L2*). This *PSUP_L2* manages the monitoring task in the second level monitoring. Therefore, in normal situation, the *CH* performs an aggregation function in which the aggregate result is accepted by *BS* without any additional communication overhead.

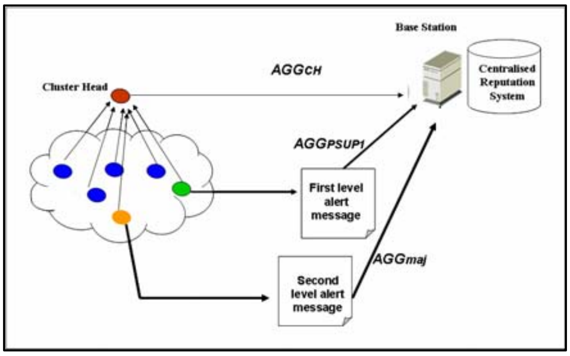

Figure 2: scheme architecture

5.2 Scheme details

The secure data aggregation scheme evolves in three regular steps and two special steps. When *CH* and *PSUP_L1* are normal, the aggregation process terminates after the first three regular steps. However, if attack on *CH* and/or *PSUP_L1* is detected, the protocol executes extra special steps 4 and/or 5, depending on attack scenario.

5.2.1 Regular steps

***1- Initialisation**:* This step includes boot setup and cluster formation. The boot setup occurs before nodes deployment, in which the *BS* assigns each sensor *i* a single identifier Id_i, and a unique symmetric encryption key K_i^{BS} which *BS* shares with the sensor *i.* In addition, we assume that a sensor can securely setup pair-wise keys with each of its neighbor nodes once deployed. The cluster formation occurs when nodes are deployed, in which sensors self-organize into disjoint cliques. Once clusters (cliques) are formed, nodes inside each cluster elect one of them as the cluster-head *(CH)* to act as aggregator. Each *CH* sends to *BS* the list of sensors in its own cluster.
The aggregation process can be done as a response to a *BS's* query. The *BS* propagates a query message to the cluster-heads. In each query, the *BS* elects dynamically a principal supervisor for first level (*PSUP_L1*) and a principal supervisor for second level (*PSUP_L2*) in each cluster. It piggybacks these two identities in query message dissemination. However, the choice of *PSUP_L1* and *PSUP_L2* is not trivial. We assume that the *BS* has the ability of reasoning about sensor behavior, by maintaining a centralised reputation system. Thus the *PSUP_L1* and *PSUP_L2* are elected among the sensors with high good reputation score. When *CH* receives query, it broadcasts it to all sensor nodes in its cluster.

***2- Data filtering and aggregation**:* Our scheme exploits the broadcast nature of radio transmission to distribute the task of aggregation over all the nodes in the cluster, i.e. all nearby nodes of each aggregator, participate in aggregation function and gather the data through passive listening. In spite of the participation of all nodes to the aggregation function, only the *CH* sends its aggregate result to the *BS*. The other nodes act as supervisors to ensure the accuracy of aggregation result and react only when this accuracy is violated. We assume that the *CH* does not have data itself.
Aggregation process is done in rounds, inside each cluster, as well as all aggregation protocols available in the literature (synchronization is required). The l^{th} aggregation round on a cluster Cl_i, led by cluster-head CH_i, is done as follows:

$$i \rightarrow * : Id_i, S_i \qquad (1)$$

Each node $i \in Cl_i$, except CH_i, broadcasts its reading S_i. Note that an attacker can not impersonate a node *i.* Indeed, communications inside a cluster are single-hop only and the messages do not go through intermediate nodes where they could potentially be corrupted maliciously. As a consequence, we do not need to use *MAC* to guarantee message integrity. However, to handle non-malicious corruptions from the environment, we use a mechanism such as *CRC* (Cyclic Redundancy Check) [Ning, 05].
Each node $x \in Cl_i$, receives (and collects) all the broadcasted messages, sent by members of cluster.

Before achieving aggregation function, we add a prior step to data aggregation model, where after receiving readings from sensor nodes, each node (including the aggregator) performs locally an analysis of the input data before aggregation, and tries to identify potentially multiple "bogus" sensor readings and removing them from the computation of the aggregate function. This prior step is very important before performing aggregation. Indeed, if the adversary upsets sensor readings by directly manipulating the environment, it will surely pervert the aggregation results. To check the reliability of data, a robust statistical technique must be applied for identifying outliers. A good outlier detection algorithm should detect most of the faults and the number of false positives must be small. RAHIM uses the median which is statistically robust to outliers [Wagner, 04]. It is rule-based and hence does not require a comparison with the estimated standard deviations (which are affected by presence of outliers) of readings to decide whether a value is an outlier or not [Kumar, 09]. For each node in the cluster, the median of the readings of neighbor nodes is calculated. If reading of the node differs from the median by more than a threshold value, it is declared as an outlier. The algorithm is defined in Algorithm 1. It is assumed that the mean and standard deviation of the measurement error (calibration error) of the sensor used on board is provided by the manufacturer. The threshold is set as twice the maximum measurement error [Kumar, 09].

Algorithm 1: Data filtering and aggregation algorithm

Input: S *set of received readings from the sensors in the cluster*
Output: aggregation result
$S_1 = \phi$
$MED = median\_of\_readings$
For each reading i *of* S *do*
    If $abs(i - MED) < threshold$ *then*
        $S_1 = S_1 \cup \{ i \}$
    EndIf
EndDo
Compute aggregation function on subset S_1

After filtering the bogus readings and calculating the aggregation function locally in each sensor node, only the *CH* sends the result (AGG_{CH}) to the *BS*.

$$CH \rightarrow BS: \\ Id_{CH},\ AGG_{CH} \parallel MAC_{K_{CH}^{BS}}(AGG_{CH}, N_a) \tag{2}$$

If there exists outlying, the *CH* includes their *Id* in the message sent to the *BS.* The message can directly transmit from *CH* to *BS* as in LEACH protocol [Handy, 02].

3- Aggregation validation: Upon receiving the message sent by a *CH*, the *BS* computes the *MAC* of the received aggregate value AGG_{CH} to check data integrity. If the *BS* does not receive an alarm within a given latency bound, it assumes that no sensor node has raised an alarm, and then concludes that the received AGG_{CH} is correct, and no malicious activity has occurred, i.e. both of first-level monitor and secondary-level monitors agree on the AGG_{CH}. The latency bound should be set according to the deployed application on the WSN.

If *BS* receives a first-level alert massage from the *PSUP_L1*, which contains an aggregation value AGG_{PSUP_L1} (calculated by *PSUP_L1*), and does not receive a second-level alert message, it concludes that the peer's monitors agree on the AGG_{PSUP_L1}. It then accepts AGG_{PSUP_L1} instead of AGG_{CH}. However, if *BS* receives a second-level alert message with the new aggregate value AGG_{maj}, it concludes that the peer monitors do not agree either on the AGG_{CH} reported by *CH* or on the AGG_{PSUP_L1} reported by the *PSUP_L1*.

Finally, the *BS* computes the total aggregation result over the partial aggregation results generated per each cluster, $AGG = f(AGG_i | \forall i, Cl_i)$.

5.2.2 Special steps

4- First-level monitoring: The Principal Supervisor (*PSUP_L1*) monitors the aggregate result (AGG_{CH}) sent by aggregator to the *BS*, in passive listening. It compares it with its own aggregate result AGG_{PSUP_L1}. In the best case, when the AGG_{CH} is correct, the *PSUP_L1* does not send any first-level alert message. This means that the *PSUP_L1* agrees on the aggregation result. However, if the *PSUP_L1* does not agree on AGG_{CH}, i.e. detects the cheating of aggregator, it raises an alert message which contains its own aggregate result AGG_{PSUP_L1}.

$$PSUP\_L1 \rightarrow BS: \\ Id_{PSUP\_L1}, AGG_{PSUP\_L1} \| MAC^{BS}_{K_{PSUP}}(AGG_{PSUP\_L1}, N_a) \quad (3)$$

Like with *CH*, if there exists outlying, the *PSUP_L1* includes their *Id* in the message sent to the *BS.*

5- Second-level monitoring: As we assume no trust in both of *CH* and *PSUP_L1*, an additional monitoring is performed by the rest of sensor nodes called *peer monitors* ($MONIT_i$). These $MONIT_i$ are responsible for monitoring the behavior of *CH* and *PSUP_L1* when sending their aggregate result to the *BS*. Without any compromising on these two cornerstone types of sensor (*CH* and *PSUP_L1*) no action is undertaken, and thus, no alert message is sent to the *BS*. However, if $MONIT_i$ detect the cheating of *PSUP_L1* or both of *CH* and *PSUP_L1*, they cooperate to generate and raise a second-level alert message to the *BS,* which contains the majority vote-based aggregate value AGG_{maj}. If we suppose that the number of $MONIT_i$ is *n*; it is not efficient to send *n* alert-messages to the *BS*. Contrary to previous protocols, we design a principal supervisor among these peer monitors called *PSUP_L2*, which collects a complaint message from each $MONIT_i$ which does not agree on aggregate result, and performs a majority vote to generate an alert message.

$$MONIT_i \rightarrow PSUP\_L2: \\ Id_{MONIT_i}, H(AGG_{MONITi}) \| MAC^{BS}_{K_{MONITi}}(AGG_{MONITi}, N_a) \quad (4)$$

Improvement: It is obvious that the second level monitoring is more expensive than the first level monitoring, because of the complaint messages transmission. However, since the aggregation result can be of any size, each $MONIT_i$ just sends $H(AGG_i)$ (hash of AGG_i) instead of AGG_i, in order to reduce the transmission overhead. Because all nodes of the cluster overhear the same sent message, all honest nodes must report the same aggregate value AGG_i. As a consequence, they will report the same hash of the aggregation result $H(AGG_i)$, assuming that they use the same hash function H. After collecting sufficient number of complaint message including AGG_i and their signature, the *PSUP_L2* computes an XOR-ed MAC over the received MACs, and sends the followings second-level alert message to the *BS*:

$$PSUP\_L2 \rightarrow BS: \\ Id_{PSUP\_L2}, AGG_{maj} \| \oplus MAC^{BS}_{K_{MONIT_j}} (AGG_{MONIT_j}, N_a) \tag{5}$$

If a node x of a cluster fails to send its computed aggregate AGG_i, the *PSUP_L2* includes Id_x in the second-level alert message sent to the *BS*, to notify that the computed XOR-ed MAC was not computed over the contribution of node x. In case of conflicting hash aggregation values (and thus, conflicting computed aggregation values), *PSUP_L2* chooses the majority voted hash aggregation value (the hash aggregation result with the highest occurrence) to be the hash of the aggregation result of the cluster $H(AGG_{maj})$. In case $H(AGG_{PSUP\_L2}) \neq H(AGG_{maj})$, *PSUP_L2* asks any sensor among the majority which reported $H(AGG_i)$, to send it back the aggregation result AGG_i. In all cases, *PSUP_L2* computes the XOR-ed MAC only over the MACs related to the majority voted hash aggregation result, and it reports the *Id* of each node whose computed aggregation value differs from the cluster aggregation result AGG_{maj}.

As we mentioned in section 3, the number of compromised sensors is less than the well-behaving sensor. Thus, the *PSUP_L2* ignores any message if it receives less than *n/2* alert messages. This means that a compromised node cannot send a complaint with an aim of compromising a correct result.

6 Security analysis

The proposed security analysis of our protocol RAHIM focuses on:

- *Resilience against false data injection attack:* Can an attacker successfully alter the aggregate result by forging bogus data reading?
- *Resilience against False aggregation attack:* Can an attacker successfully mislead the *BS* to accept a false aggregation result by tampering with aggregation process?
- *Resilience against data rejection:* Can availability be well considered even when subversive activities persist?
- *Resilience to failure aggregator:* Can the protocol ensure the accuracy of aggregate result in the case of aggregator failure?

6.1 Resilience against false data injection attack

The *false data* injection attack occurs when an attacker modifies data reading reported by nodes under its direct control [Can, 06]. It is very difficult to detect such attack.

However, the majority of the existing solutions to secure data aggregation assume that the sensor nodes are reporting data truthfully [Labraoui, 09] or accept only data reading that is bounded between minimum and maximum values, according to the application [Bagaa, 07]. However, this later technique reduces the impact of false data injection attack but makes it very difficult to differentiate between emergency events sent by good nodes and malicious events. Other protocols which rely on concept of trust are emerged recently. They are socially inspired and use the new paradigm of reputation inherited from human behavior to isolate the injection of bogus data [Maarouf, 09], [Kumar, 09], [Junbeom, 05]. Nevertheless these approaches are prone to bad mouthing attack in which a compromised node can falsely accuse well-behaving nodes of malicious actions or falsely praise bad-behaving nodes. In addition, an extra transmission overhead is generated by the periodic exchange of reputation values between the nodes. In our protocol, we cope with the false data injection attack in a lightweight manner by adding a prior step to data aggregation model, in which data filtering algorithm is performed locally before computing aggregation function.

To prove the effectiveness of the data filtering algorithm based on the median, we have tested it in a simulation environment using Matlab.

Imagine the scenario of typical temperature-collection application: A group of sensors such as Micas have been deployed to collect temperature samples. Suppose each group of *n* nodes organized themselves into a cluster. They take temperature measurements every minute and send these measurements to the cluster-head. It is clear that sensor readings like temperatures can be highly correlated in a small geographical area. This correlation among sample elements is a naturally existing phenomenon.

The sample was generated by the *randn* function. The Peak Attacker was simulated by a function that replaces those sample elements to a common value, which correspond to the proportion determined by *k*. This replacement was done in the wide surroundings of the real expected value of the sample. To obtain the maximum distortion reachable by the Peak Attacker, we have made 50 simulation runs for different values of *k* (i.e., different proportion of compromised nodes). Figure 10 shows the error deviation of median calculations for typical temperature-collection application. The error deviation is very insignificant below of 50 percent of compromised nodes. But for higher *k* values, the results of the median calculation rapidly decline. In Figure 11, we remark that the aggregation value after filtering bogus data is very close to the real average of the original sample. In both Figures 10 and 11, the median has a breakdown point of 50. In conclusion, simulation results of false data injection attack show that the median calculation incurs only a small computation overhead and still produces precise estimates for 50 percent of compromised nodes. The median is then a robust statistical method in presence of several bogus data (outliers) and produces zero false positives below this threshold. Thus, our secure aggregation scheme is immune against false data injection attack.

6.2 Resilience against false aggregation attack

Because aggregator is a cornerstone in data aggregation process, and compromising it, lead to the attack success; it is very important to verify the correct behavior of aggregator nodes. For this reason we use a monitoring-based approach to ensure the accuracy of aggregation result. However, because no trust is supposed in any sensor in the cluster, several attack scenarios can occur. We explain them in the following section.

- Compromised cluster-head attacks. If the *CH* is compromised, it can forge arbitrary aggregation results and generate matched MAC of these false results. In our protocol, such attacks will be effectively defended, since we introduce a first-level monitoring. The *PSUP_L1* raises alert against the cluster-head's false aggregate result, and provides the *BS* with its own aggregate result.

- Selective attack on principal supervisor of first-level. An obvious idea of the attacker is to compromise both the *CH* and the *PSUP_L1* together. However, in our scheme, such attacks will also be defended because we introduce the second-level monitoring in which *PSUP_L2* raises an alert on the basis of received complaint messages and provides correct result to *BS*.

- Compromised principal supervisor of second-level. If the *PSUP_L2* is compromised, it tries to fabricate an alert message to mislead the *BS* to accept its own aggregate result instead of the real value. However, the *PSUP_L2* can not forge the legal MAC to generate a majority vote, and thus it can not generate a valid alert message.

6.3 Resilience against data rejection

Data rejection is an important problem of secure aggregation protocols. A protocol suffering from this kind of problem can not prevent a bogus data from infecting the global aggregation, leading in cancellation of all steps of aggregation process. Our scheme RAHIM overcomes the total rejection by stopping locally invalid data during the aggregation phase (by data filtering algorithm) and by relying on concept of monitoring. The role of theses monitors is to provide a valid aggregation value to the *BS*, avoiding the data rejection when data integrity does not hold. Thus our scheme ensures more availability than other proposals.

6.4 Resilience to aggregator failure

Because the task of data aggregation is distributed to all sensors in the cluster, and our network model is based on the use of cliques, it is more tolerant to aggregator nodes failures than other protocols like [Du, 03], [Hu, 03] and [Przydatek, 03]. Since all nodes of the cluster compute the aggregation result, if a *CH* failure happens during the aggregation process; our framework can be adapted to recover from the failure and continues the aggregation from the point of failure.

7 Performance evaluations

The rationale to use RAHIM is to conserve energy by requiring no cryptographic operations and no overhead transmission when sensor nodes behave correctly. This rationale is legitimate only if RAHIM does not incur much larger energy cost of data transmission than other aggregation protocols, and if energy cost of monitoring with RAHIM in the long run is lower than the energy cost of cryptographic operations. In following section, we demonstrate that the two conditions are verified for RAHIM.

7.1 Transmission overhead

The main purpose of conducting aggregation is to reduce communication overhead. But security mechanisms have some extra overhead. Our secure aggregation scheme attempted to maintain this purpose by introducing lower transmission overhead, while providing maximum security level without any degradation. Relying on two hierarchical levels of monitoring, the density of peer monitoring nodes does not increase contention to access the medium. The scheme is then independent to the size of network contrary to work in [Cristofaro, 09] and [Du, 03]. One advantage of the assumed network model is the cluster formation based on *Sun et al*. protocol that reduces the overhead because periodic *CH* election inside a cluster does not change the cluster sensor members. Whereas in other approaches like LEACH [Handy, 02], TEEN [Manjeshwar, 01] and APTEEN [Manjeshwar, 02], where the *CHs* are first elected then clusters are formed, a periodic *CH* election implies new formed clusters, and consequently extra energy consummation due to the exchanged messages.

To be convenient for analysis and comparison, we assume that, in each transmitted message, the length of the data, node *Id* and MAC are of little difference in most protocols. We take the number of transmitted messages as our metric for communication overhead. We consider an ideal transmission in cluster with n sensor nodes, which report their reading. For the second step, each sensor node sends its reading to the *CH*. We use m to represent the length of the data reading, c for the length of the node *Id* plus MAC, w for the length of *node Id* plus CRC, and p for the length of hash value plus MAC, with $w < c$. In the next step, each *CH* retransmits the MAC of the aggregate value. The aggregation function output has the same length as the original sensor reading. Different scenarios of attacks are detailed below.

Scenario 1: When the sensor nodes behave correctly, i.e. without any attack. The total number of bits transmitted in aggregation process is $(n+1)m+nw+c$. For comparison, with insecure aggregation method (TAG), n messages are aggregated into 1 message at each aggregator node, so each node only needs to transmit $m+w$ bits. This requires transmission of $(n+1)m+(n+1)w$ total bits. Our secure aggregation involves only the data aggregation phase and does not require any additional messages. Compared with the insecure aggregation, our mechanism has only an overhead of 4 bytes.

Scenario 2: If only the aggregator is compromised in the cluster, then step four is executed. In this case, our scheme generates only one additional message of $c+m$ bits to the aggregation process. So the total number of bits transmitted is $(n+2)m+nw+2c$. This is a very insignificant transmission overhead compared with other schemes reaction in presence of compromised aggregator.

Scenario 3: When *PSUP_L1* is compromised and *CH* is honest, the step five is executed. This is the worst case in which the total overhead generated is equal to $(n+3)m+nw+tp+3c$. t represents the number of honest nodes that generate complaint message and $t < n$.

Scenario 4*:* In colluding attack, when both of the *PSUP_L1* and *CH* are compromised, the *PSUP_L1* does no generate an alert message against aggregator colluding with it. The overhead is equal to $(n+2)m+nw+tp+2c$.

According to Hu and Evans [Hu, 03], the total number of bits generated by its protocol with b^d leaf nodes is:

$$m(2b^{d+1}-b^2-b)/(b-1)+c(2b^{d+1}+b^d-b^2-2b)/(b-1) .$$

Where the leaves are d hops away from the *BS* and each node has b children. To give a sense of what these numbers mean for typical applications, we select m=22 bytes, c=14 bytes, w=10 bytes and $p=22$ bytes, based on the assumptions in [Perrig, 01] (for messages where no MAC is included, 2 bytes are required for a message integrity CRC). Given a network with n=16 (b=4 and d=2), the total communication in a time segment where each sensor node transmits a reading is 544 bytes with unsecure aggregation and 1352 bytes in Hu's protocol. However, in our framework the total communication overhead is 548 bytes in scenario 1, 584 bytes in scenario 2, 1060 bytes in scenario 3 and 1024 bytes in scenario 4, assuming that number of honest nodes is t=10 (40% of compromised node). In summary, through analysis and comparison, as shown in Table 2, we can see that our protocol does not add much communication overhead to pure aggregation without security. Meanwhile, compared with Hu's secure aggregation protocol, in which the overhead increases in an exponential way, our protocol provides much security, but with lower communication overhead.

Leaf Nodes		**16**	**32**	**64**	**128**
TAG		4.3 KB	8.4 KB	16.6 KB	33 KB
Hu and Evans [Hu, 03]		10.8 KB	38.4 KB	49.4 KB	159.8 KB
Our Scheme	**Scenario1**	4.3 KB	8.4 KB	16.6 KB	33 KB
	Scenario2	4.6 KB	8.7 KB	16.9 KB	33.3 KB
	Scenario3	8.4 KB	12.5 KB	24.1 KB	47.1 KB
	Scenario4	8.1 KB	12.2 KB	23.8 KB	46.8 KB

Table 2: Transmission overhead comparison with 40% of compromised nodes.

7.2 Computational overhead

Cryptography causes considerable extra consumption of energy, mainly due to packet overhead, which leads consequently to a shorter network lifetime [Perrig, 01], [Karlof, 04]. Including energy consumed on CPU processing, every cryptographic primitive requires a different amount of time and a different number of CPU cycles for execution, resulting in different energy consumption values. For example, Skipjack requires 22,044.60 CPU cycles and consumes 71.76 µjoules for calculating a 29-byte packet MAC [Wander, 05]. However, the majority of previous protocols address the integrity of data aggregation in wireless sensor networks by relying on cryptographic operation as endorsement proof. Each sensor reports its reading with its MAC, and sends it to the aggregator. Consequently, we note that both of [Cristofaro, 09] and [Du, 03] induce a high transmission and computational overhead neglecting energy cost even in no attack existence. Contrary to these proposals, our scheme relies on cheat proof instead of endorsement proof. By this fact, all the sensor nodes in cluster except cluster-head, act as monitors during the aggregation process. In normal situation, we do not need to use the MAC to guarantee message integrity when sensors broadcast their reading, because all communications are single hop, and the messages do not go through intermediate nodes where they could potentially be corrupted maliciously. However, only *CH* calculates the MAC and sends it with the aggregate result. Doing so, we avoid some number of CPU cycles for execution. We also avoid adding additional bytes to the original message, and save on energy that would be spent sending these bytes.

7.3 Energy cost of monitoring

Overhearing is often considered a cause of energy wastage [Iima, 09]. However, the peer monitors do not need to listen during long periods. They only listen during the aggregation process, which is done in round as a response to *BS*'s query. The assumed structure of cluster based on single-hop communication between sensors, fully takes advantage of the broadcast feature of radio channels and thus no extra energy is required for receiving messages if the sensor is set to promiscuous listening mode. This is the same as the watchdog mechanism [Maarouf, 09]. On other hand, our proposal mitigates the burden of monitoring cost on energy-constrained sensors by discharging them from systematic computing some proof based on cryptographic primitives imposed by checking integrity. On the other hand, peer monitors are dedicated to compute a simple aggregation function like max, min and mean. As reported in [Wu, 06], the number of basic operation in min/max and mean functions is equal to 23 operations against 4192 operations in RC5 with 16-byte packet. It is obvious that aggregation operations are much simpler than cryptographic operations.

7.4 Comparison of Security Features

In Table 3 we summarize the security features of our proposal compared with other relevant algorithms present in the literature. The feature aggregation type indicates who is responsible for the aggregation: hop-by-hop means that multiple aggregators' model is used in which each node adds its own value to the aggregate while CH means that the local aggregation is performed by the cluster head. However, in *SIA* protocol, a single aggregator model is used in which all individual data in the WSN travels to only one aggregator point in the network before reaching the base station. The feature insider attack resilience indicates the resilience against the bogus data injection, i.e. when attacker manipulates the sensing data. We can show that all the previous solutions do not handle this type of attack. Table 3 also indicates if the protocol is resilient against malicious aggregators and aggregators' failures in columns 4 and 5 respectively. The column 6 indicates the resilience against data rejection that is the main drawback of almost all existing solutions that focus on integrity of data aggregation. The last column denotes the management policy of protocols. By unique rule, we refer to the systematic use of cryptographic primitives even when no attack existence. By adaptive rule, we refer to the adaptive reaction according to the attack scenario. In this case cryptographic primitives are used only when necessary, i.e. only when malicious activities are detected.

	Aggregation type	Insider attack resilience	Malicious aggregator resilience	Aggregator failure resilience	Data rejection resilience	Management policy
SDA [Hu, 03]	Hop-by-hop	No	Yes	No	No	Unique
SIA [Przydatek, 03]	Unique aggregator	No	Yes	No	No	Unique
WDA [Du, 03]	Hop-by-hop	No	Yes	No	No	Unique
SDAP [Yang, 06]	Hop-by-hop	No	yes	No	No	Unique
FAIR [Cristofaro,09]	Hop-by-hop	No	yes	Yes	Yes	Unique
RAHIM Our solution	CH	Yes	Yes	Yes	Yes	Adaptive

Table 3: data aggregation protocols: comparing the security features.

7.5 Simulation results

In this section, we perform simulation study to further demonstrate the feasibility and the effectiveness of our secure aggregation scheme. We evaluate how our scheme performs in terms of latency, aggregation accuracy and energy efficiency. The protocol is implemented in *network simulator* – NS2 [NS2]. Mannasim framework [Mannasim] was used to introduce new modules for design, development and analysis of different WSN applications in NS2.

We have used the Skipjack algorithm for computing MACs. The channel capacity is assumed to be constant and equal to 10 Kbps over the wireless link and ideal channel have been considered. The sensor nodes were deployed in 100 meters by 100 meters area. Because our scheme is running in each cluster, we carry out the simulation in a cluster and we varied the number of sensor nodes from 6 to 36 to change cluster density. The transmission range for each sensor node is 40 m. Table 4 summarizes the parameters for the simulation of Crossbow mica2 sensor node. Transmit Power (Pt_) is the power with which the signal is transmitted. The Transmit Power (Pt_) decides the transmission range for the sensor node. Transmit Power (txPower) is the power consumed by the transceiver to transmit a data packet. Receive Power (rxPower) is the power consumed to receive a data packet.

Parameter	**Value**
Number of nodes in a cluster	6, 16, 26 and 36
Number of rounds	10
Transmit Power (Pt_)	8.564E-4 mW
Transmit Power (txPower)	0.036 mW
Receive Power (rxPower)	0.024 mW
Initial energy	10 J
Coverage area	100m x 100m
Transmission range	40 m

Table 4: Simulation Parameters.

The simulation was run using different scenarios of attacks and 40% of compromised nodes are inserted in the cluster. Ten queries are initiated by the base station. The simulation results were obtained by calculating the average of all runs.
For comparison purpose, we also implement the insecure aggregation protocol (TAG) and classical secure aggregation scheme in which integrity violation induces a data rejection.

1. Latency: We mean by *latency*, the average delay between the *BS* request and the delivery of aggregate result to the *BS* from the leaf nodes. *Figure 3* illustrates the benefit of using the monitoring mechanism to provide a correct result to *BS* without referring to cancel aggregation process when cheating is detected. Comparing with insecure aggregation (TAG), the delivery speed in our scheme is *constant* and *very close* to TAG in both scenario1 and scenario2. However, in scenario3 and scenario4, this delay increases relatively when number of nodes increases, since it will require sending complaint messages.

2. Accuracy: In ideal situations when there are no compromised nodes in the network, RAHIM should get 100% accurate aggregation results. However, because the sensors are deployed in untrusted environment, and can be compromised, the aggregation accuracy is affected. We define the accuracy metric for the average function as the ratio between the collected average by the data aggregation scheme used and the real average of all individual sensor nodes. A higher accuracy value means the collected average using the specific aggregation scheme is more accurate. An accuracy value of 1.0 represents the ideal situation.

Figure 4 shows the accuracy of TAG and RAHIM from our simulation in which we consider a cluster with 26 nodes. Here we observe that the accuracy decreases as the proportion of compromised nodes increases in the insecure aggregation scheme TAG which is very sensitive in untrusted environment. Whereas, in RAHIM, the accuracy is very high below 50% of compromised nodes. Thus, in all attack scenarios, RAHIM has better accuracy than TAG.

3. Energy efficiency: RAHIM uses monitoring mechanism to protect integrity of data aggregation. By this mechanism, alert messages are raised when cheating is detected. This introduces energy consumption. Hence, in order to investigate energy efficiency of our scheme, we first study the residual energy of our proposed scheme. Secondly, we study the energy saving of RAHIM compared to classical secure aggregation scheme.

3.1 Residual energy: We analyze the average of *Residual energy* while varying the number of sensors in the cluster in the fourth attack scenarios. *Figure 5(a)* and *5(b)* shows the effect of increasing the number of nodes on the average residual energy in one round. Initially each node has 10 joules. We remark in *Figure 5(a)* that the power consumption of our proposal is *very close* to TAG in normal situation (without attack). However in presence of attack, our scheme adapts its reaction in function of attack scenario and does not require much energy than TAG. Thus, our secure aggregation scheme maintains the purpose of aggregation in term of energy efficiency.

3.2- Energy gain: In our scheme, when bogus aggregation result is sent, *BS* does not cancel the aggregation process because it is supplied by correct result piggybacked in alert message. In this metric, we analyze the impact of data rejection on the energy consumption while varying the number of data rejection. We simulate a classical secure aggregation scheme in which aggregation process is cancelled and then all steps are re-run. *Figure 6* depicts clearly the energy spent with one, two and three data rejections. However, *Figure 7*, *8* and *9*, illustrate the energy saving by RAHIM compared to classical scheme respectively with one, two and three data rejections.

In summary, our scheme RAHIM significantly outperforms classical secure aggregation scheme in term of energy consumption under attack scenarios.

8 Conclusions

We have presented RAHIM a novel secure data aggregation scheme in WSN that enforces both availability and accuracy of the data aggregation. The proposed scheme is based on a novel application of adaptive hierarchical level of monitoring providing accuracy of data aggregation result in lightweight manner, even if all aggregator nodes and a part of sensors are compromised in the network. Contrary to previous proposals, our scheme relies on cheat proof instead of endorsement proof mechanism. Enabling cryptography is directly related to the accuracy of aggregate result. When accuracy is violated, security is turned on immediately and monitors play their role efficiently supplying the *BS* by correct aggregate value. This avoids a high cost interactive verification phase. Moreover, in normal situation, i.e. without any attack, our scheme involves only the data aggregation phase and does not require any additional transmission overhead. In addition, RAHIM is robust against bogus data injection and total data rejection and has the ability to recover from aggregator failure without neglecting energy efficient, providing thus much higher availability than other security protocols.

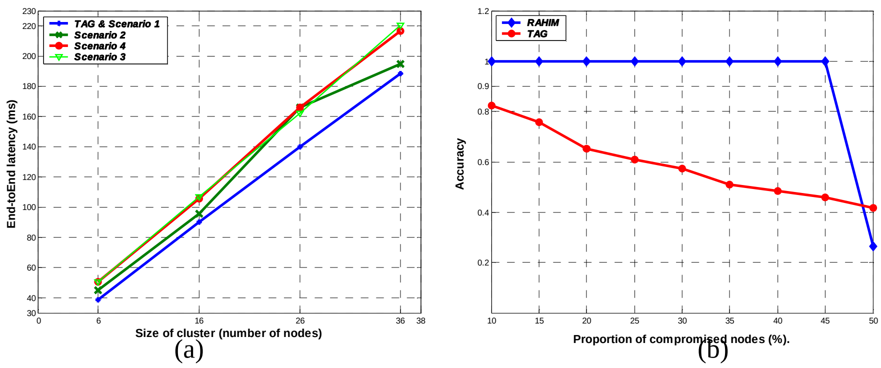

Figure 5: (a) Residual energy in normal situation. (b) Residual energy in presence of attack.

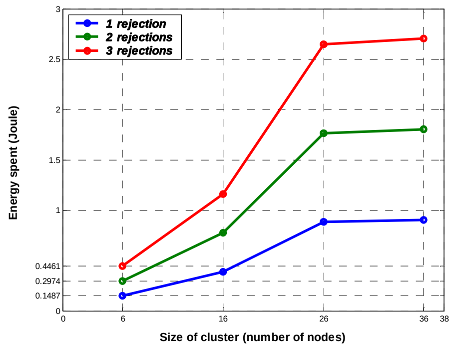

Figure 6: Energy spent with data rejection.

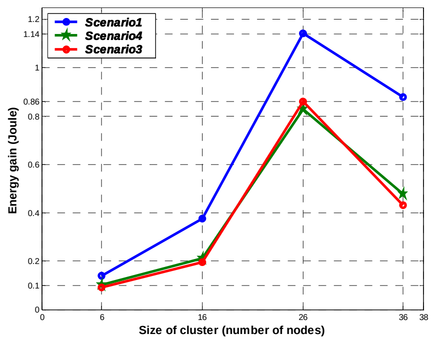

Figure 7: Energy gain of RAHIM (1 rejection).

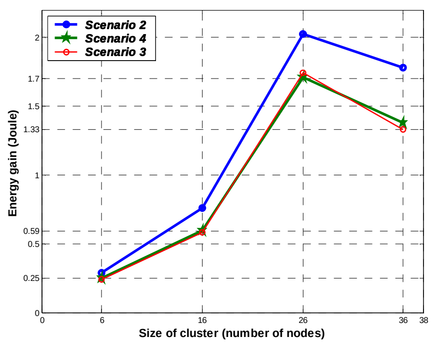

Figure 8: Energy gain of RAHIM (2 rejections).

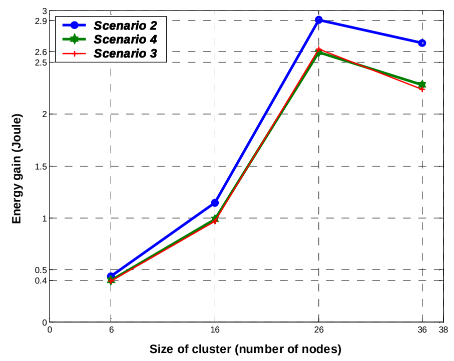

Figure 9: Energy gain of RAHIM (3 rejections).

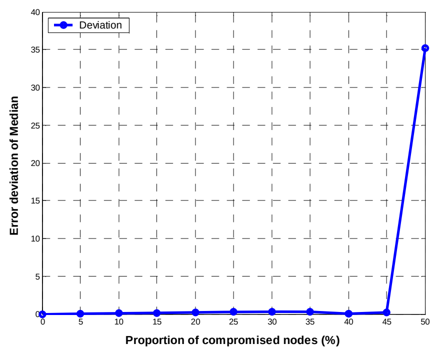

Figure 10: Error deviation of Median vs Proportion of compromised nodes in a cluster

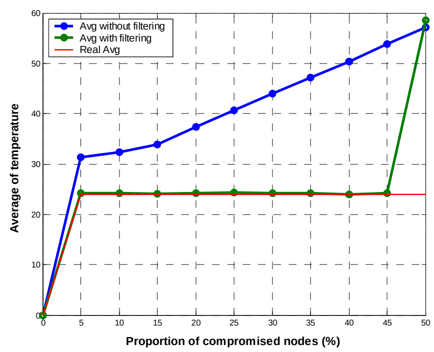

Figure 11:Comparison of average aggregation vs Proportion of compromised nodes in a cluster

References

[Akyildiz, 02] I.F., Su W., Sankarasubramaniam Y., Cayirci E.: A Survey on Sensor Networks, IEEE communications Magazine, 2002, 40(8); 102–114.

[Bagaa, 07] Bagaa M., Lasla N., Ouadjaout A., Challal Y.: SEDAN: Secure and Efficient Protocol for Data Aggregation in Wireless Sensor Networks, 32nd IEEE Conference on Local Computer Networks (LCN), 2007, pp. 1053 - 1060.

[Can, 06] Can H., Perrig A., Song D.: Secure Hierarchical In-network Aggregation in Sensor Networks, 13th ACM Conference on Computer and Communications Security, 2006, pp. 278–287.

[Corporation, 07] Corporation C.: Mica2 datasheet, Reviewed 10th of October 2007, http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICA2_Datasheet.pdf.

[Cristofaro, 09] Cristofaro E. D, Bohli J. M., Westhoff D.: FAIR: Fuzzy-based Aggregation Providing in-network Resilience for Real-time Wireless Sensor Networks, 2nd ACM Conference on Wireless Network Security (WiSec), 2009.

[Du, 03] Du W., Deng J., Han Y. S., Varshney P.: A Witness-Based Approach For Data Fusion Assurance In Wireless Sensor Networks, GLOBECOM'03, 2003, volume 3, pp. 1435-1439.

[Handy, 02] Handy M., Haase M., Timmermann D.: Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection, IEEE Mobile and Wireless Communications Networks, 2002, Stockholm.

[Hu, 03] Hu L. and Evans D.: Secure Aggregation for Wireless Networks, Symposium on Applications and the Internet Workshops, 2003.

[Iima, 09] Iima Y., Kanzaki A., Hara T. and Nishio S.: An Evaluation of Overhearing-Based Data Transmission Reduction in Wireless Sensor Networks, 9th International Conference on Mobile Data Management: Systems, Services and Middleware, 2009, pp. 519-524.

[Junbeom, 05] Junbeom J.H., Yoonho L., Seongmin H., Hyunsoo Y., Trust-based aggregation in wireless sensor networks. International Conference on Computing, Communications and Control Technologies, July 2005.

[Karlof, 04] Karlof C., Sastry N., Wagner D.: TinySec: A Link Layer Security Architecture for Wireless Sensor Networks, 2nd International Conference on Embedded Networked Sensor Systems, 2004, Baltimore, MD, USA, pages 162-175.

[Kumar, 09] Kumar A., Ribeiro V. J.: REEF: a reliable and energy efficient framework for wireless sensor networks, 1st International Communication Systems and Networks and Workshops (COMSNETS), 2009.

[Labraoui, 09] Labraoui N., Guerroui M., Aliouat M., Zia T.: Data Aggregation Security Challenge in Wireless Sensor Networks: A Survey, Ad hoc & Sensor Networks, International Journal, 2011, vol. 12, no. 3-4, pp. 295-324.

[Maarouf, 09] Maarouf I. , Baroudi U. and Naseer A. R.: Efficient monitoring approach for reputation system-based trust-aware routing in wireless sensor networks, IET communications, 2009, vol. 3, issue 5, pp. 846-858.

[Mannasim] Mannasim framework, <http://www.mannasim.dcc.ufmg.br>.

[Manjeshwar, 01] Manjeshwar A., Agrawal D.: TEEN: A Protocol for Enhanced Efficiency in WSN, IEEE International Symposium on Parallel and Distributed Processing, 2001, San Francisco, USA, pp. 2009–2015.

[Manjeshwar, 02] Manjeshwar A., Agrawal D.: APTEEN: A Hybrid Protocol for Efficient Routing and a Comprehensive Information Retrieval in WSN, IEEE International Symposium on Parallel and Distributed Processing, 2002, Florida, USA, pp. 195–202.

[Ning, 05] Ning P., Sun K.: How to Misuse AODV: A Case Study of Insider Attacks against Mobile Ad-hoc Routing Protocols, Ad Hoc Networks, 2005, Vol. 3, No. 6, pp. 795-819.

[NS2] Network Simulator, <http://www.isi.edu/nsnam/ns>.

[Pai, 10] Pai H.T. , Deng J. and Han Y. S.: Time-slotted voting mechanism for fusion data assurance in wireless sensor networks under stealthy attacks, Computer communications, 2010, vol. 33, n°13, pp. 1524-1530.

[Perrig, 04] Perrig A., Stankovic J., Wagner D.: Security in Wireless Sensor Networks, Communication of the ACM, June 2004.

[Perrig, 01] Perrig A., Szewczyk R., Wen V., Cullar D., Tygar J. D.: Spins: Security Protocols for Sensor Networks, 7th Annual ACM/IEEE International Conference on Mobile Computing and Networking, 2001, Rome, Italy, pp. 189-199.

[Przydatek, 03] Przydatek B., Song D., Perrig A.: SIA: Secure information aggregation in sensor networks, 1st ACM International Conference on Embedded Networked Sensor Systems (SenSys'03), 2003.

[Sun, 06] Sun K., Peng P., Ning P., Wang C.: Secure Distributed Cluster Formation in Wireless Sensor Networks, 22nd Annual Computer Security Applications Conference, December 2006.

[Wagner, 04] Wagner D.: Resilient Aggregation in Sensor Networks, ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN '04), 2004.

[Wander, 05] Wander A. S., Gura N., Eberle H;, Gupta V., Shantz S.: Energy Analysis of Public-Key Cryptography on Small Wireless Devices, IEEE PerCom, 2005.

[Wu, 06] Wu K., Dreef D., Sun B., Xiao Y.: Secure Data Aggregation without Persistent Cryptographic Operations in Wireless Sensor Networks, 21st IEEE International Conference on Performance Computing and Communications, 2006, pp. 85.

[Yang, 06] Yang Y., Wang X., Zhu S., Cao G.: SDAP: A Secure Hop-by-Hop Data Aggregation Protocol for Sensor Networks, 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MOBIHOC' 06), 2006.