Telecommun Syst (2013) 54:3–17
DOI 10.1007/s11235-013-9712-3

Reactive and adaptive monitoring to secure aggregation in wireless sensor networks

Nabila Labraoui · Mourad Gueroui ·
Makhlouf Aliouat · Jonathan Petit

Published online: 12 July 2013

Abstract Data aggregation is considered as one of the fundamental distributed data processing procedures for saving the energy and minimizing the medium access layer contention in wireless sensor networks. However, sensor networks are likely to be deployed in an untrusted environment, which make them vulnerable against several attacks. A compromised node may forge arbitrary aggregation value and mislead the base station into trusting a false reading. Secure in-network aggregation can detect such manipulation. But, as long as such subversive activity is, reliable aggregation result can not be obtained. In contrast, the collection of individual sensor node values is robust and solves the problem of availability, but in an inefficient way. Our work seeks to bridge this gap in secure data collection. We propose a framework that enhances availability with efficiency close to that of in-network aggregation avoiding over-reliance on sensors. To achieve this, we design a scheme that is built on one core concept: no trust is supposed in any sensor. Therefore, we design a two hierarchical levels of monitoring to ensure the integrity and the accuracy of aggregate result, only when necessary, i.e. only when malicious activities are detected. Relying on this new type of monitoring mechanism, the framework has the ability to recover from aggregator failure without neglecting energy efficiency, providing thus much higher availability than other security protocols.

Keywords Data aggregation · Hierarchical monitoring mechanism · Wireless sensor networks · Security

1 Introduction

Wireless sensor networks (WSNs) are becoming more and more spread and both industry and academia are focusing their research efforts in order to improve their applications [1]: battlefield surveillance, target tracking, environmental and health care monitoring, fire detection, and traffic regulation. However, sensor networks have extremely constrained resources like energy, bandwidth and capabilities of processing and storing data. Therefore, the key challenge in sensor networks is to maximize the lifetime of sensor nodes due to the fact that it is not feasible to replace the batteries of thousands of sensor nodes. Data aggregation is considered as one of the fundamental distributed data processing procedures for saving the energy and minimizing the medium access layer contention in wireless sensor networks [2]. Data aggregation is presented as an important pattern for routing in the wireless sensor networks. The basic idea is to merge the data from various sources, reroute it with the elimination of the redundancy, and thus, reducing the number of transmissions and saving the energy [3]. Subsequently, data aggregation attracted a great deal of attention and there are extensive work on data aggregation schemes in sensor networks [3–8]. Interested readers may refer to [9] and [10] for surveys on this topic. These efforts share the assumption that all sensors are trusted, and all communications are secured. However, in reality, sensor networks are likely to be deployed in an untrusted environment, which make them

N. Labraoui (✉)
STIC University of Tlemcen, Tlemcen, Algeria
e-mail: labraouinabila@yahoo.fr

M. Gueroui
PRISM University of Versailles, Versailles, France

M. Aliouat
University of Setif, Setif, Algeria

J. Petit
Disparaitre and Embedded Security Group, University of Twente,
Enschede, The Netherlands

vulnerable against physical node capture attacks in which intruders take control of one or more sensor nodes to subvert network's performance [11]. Capture of a 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. [12, 13]. Considering the data aggregation scenario, compromised nodes can successfully forge authenticated false reports to their neighbours, which have no way to distinguish bogus data from legitimate ones [14]. This type of attacker is called *insider attack* [12]. 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.

Hence, data aggregation protocols must be able to function securely in the presence of possible compromised nodes within the network. Many innovative and intuitive secure aggregation schemes have been proposed for solving the problem of security in sensor networks. A survey of these works is presented in [15], theses solutions fall into two main categories: cryptography-based schemes and reputation-based schemes. Both of them consider the security issue in different point of view and focus on different security requirements. In the first category, the protocols rely on pure cryptography methods to ensure the confidentiality, authentication and integrity of data. Methods, such as encryption or authentication, have the ability to verify the correctness and the integrity of an operation. But, they could not eliminate all attacks and especially the insider attacks. In the second category, reputation-based protocols have been proposed as an attractive complement to cryptography in securing WSNs. They rely on the concept of trust, which is socially inspired and use the paradigm of reputation inherited from human behaviour. These techniques provide the ability to detect and isolate both faulty and malicious nodes that behave inappropriately in the context of the specific WSN [16].

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 and computation overheads. For example, in cryptographic-based techniques, *proactive defence* is used in which all the nodes in the network cooperate to secure aggregation and rely on endorsement proof mechanism. To prove the validity of the aggregation result, the aggregator has to provide cryptographic primitives-based proofs from several nodes or witnesses. Then, it has to forward the proofs to the base station by piggybacking them with the aggregation result. This introduces some additional bandwidth consumption. Moreover, 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 [15]. The overhead is also increased because of the interactive verification phase imposed by checking integrity and accuracy between the base station (*BS*) and the sensors [17–22]. While, in reputation-based techniques, a *semi-proactive defence* is used in which each node acts as a monitor and maintains a reputation rating for each other node that it interacts with. This reputation rating is then used to alleviate the contribution of faulty nodes in the final aggregated data. To accelerate the building of reputation over time, nodes share their observations about other nodes with the rest of the network. However, this periodic exchange of reputation values between the nodes induces an extra transmission overhead [15].

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. For example, in cryptographic-based techniques a *Message Authentication Code* (MAC) is used in order to maintain the integrity of the data packet. The sink can detect any changes performed by the aggregator including the verification information, by checking the MAC value using its shared key. If a modification of the data packet is detected, then it will be discarded. Thus, an important amount of correct data is lost, resulting in wasting precious network resource.

From the above observations, we can notice the importance of a *reactive defence* instead of *proactive defence*. In other terms, security measurements must be used only when attack occurs. In this paper, we present a new framework called RAMA (Reactive and Adaptive Monitoring to secure Aggregation) for solving the above problems, improving reliability and ensuring 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 in reactive manner and 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. Section 2 introduces the related work. Network assumptions and threat model are presented in Sect. 3. In Sect. 4, the design goals of RAMA are presented, and Sect. 5 details our secure aggregation scheme. Section 6 provides a security analysis and Sect. 7 provides a performance evaluation. We conclude our work in Sect. 8.

2 Related works

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

Hu and Evans [17] proposed an aggregation protocol 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. [18] 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. [19] proposed a witness-based data aggregation scheme (WDA) for WSNs to ensure 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 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. [20] 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 [21], the authors propose Fuzzy-based framework (FAIR) for resilient data aggregation in real-time responsive wireless sensor networks supporting in-network processing. Like in Du's protocol [19], and in order to ensure the integrity of data during aggregation, witness nodes are often employed to confirm the result of aggregator nodes. However, witnesses do not only confirm the aggregator's result, but 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.

Jaydip [22] proposed an efficient aggregation protocol for WSNs (RSAP) that is secure and robust against malicious insider attack. The main attack considered is the injection of malicious data in the network by an adversary who has compromised a sensor's sensed value by subjecting it to unusual temperature, lighting or other spoofed environment conditions. In this algorithm, each node, instead of unicasting its sensed information to its parent, broadcasts its estimate to its neighbourhood. This makes the protocol more fault-tolerant and increases the information availability in the network. RSAP is similar to the one suggested in [23]. Author has extended the distributed estimation algorithm of [23] to make it secure and robust in presence of compromised and faulty nodes in a WSN. This proposal is topology-free because there is no need to establish and maintain a hierarchical relationship among the nodes in the network. This makes the algorithm particularly suitable for multiple users, mobile users, faulty nodes and transient network partition situations. In addition, RSAP has a very high detection rate with very low false positive and false negative rate. However, the main drawback of this scheme is the high communication overhead induced when a node is suspected to have been compromised. For example, if a node A suspects a received estimates from node B, it sends a broadcast message to each of its neighbours requesting for the value of their estimates to perform a majority vote to make sure that the suspected node B is malicious. According to the presented simulation results [22], the additional transmission and reception of messages induce an average increase of 105.4 % energy consumption in the nodes in the network. Hence, RSAP is not suitable for the dense networks.

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, investigating this crucial problem that causes waste of precious network resource motivates our work.

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, $1 \leq i \leq n$. The network is divided into clusters, each of which has a cluster-head (CH). According to Sun et al. [24] cluster formation protocol, inside each cluster (clique), each node is in the communication range of the remaining nodes of the cluster. Consequently, communication between each sensor is single-hop within a cluster. 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 in watchdog [12]. For routing purpose, we suppose that the set of CHs self-organize into multi-hop routing backbone, so that CHs far from the BS can reach the BS with the minimum spent energy and receive BS's requests. Note that the result of aggregation of each cluster is sent to the BS, without being aggregated again by other aggregators.

Like the LEACH protocol, we suppose that nodes of a cluster periodically report their readings, using a TDMA scheduling established by the CH after clusters are formed. The CH divides the time into frames, and during each frame each node of the cluster has one reserved slot: a broadcast slot, in which a node broadcasts its reading in the cluster. TDMA protocols are more power efficient since nodes in the network can enter inactive states until their allocated time slots. They also eliminate collisions and bound the delay [25].

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 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.

Table 1 Notation

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
S_i	The data reading of the sensor
Cl_{CH_i}	The cluster CL headed by the cluster-head CH_i

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 what Przydatek et al. [18] called a *stealthy attack*, 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 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.

4 Design goals

Under the aforementioned conditions, a security concept is required to reduce the overhead of the aggregation alteration due to node compromise. 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 deviates 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 secure scheme: RAMA

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

5.1 Overview of the proposed scheme

The design of RAMA 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 a 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 *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.

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. Figures 1 and 2 depicts the flowchart of the proposed scheme.

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 set up 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.

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 centralized 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

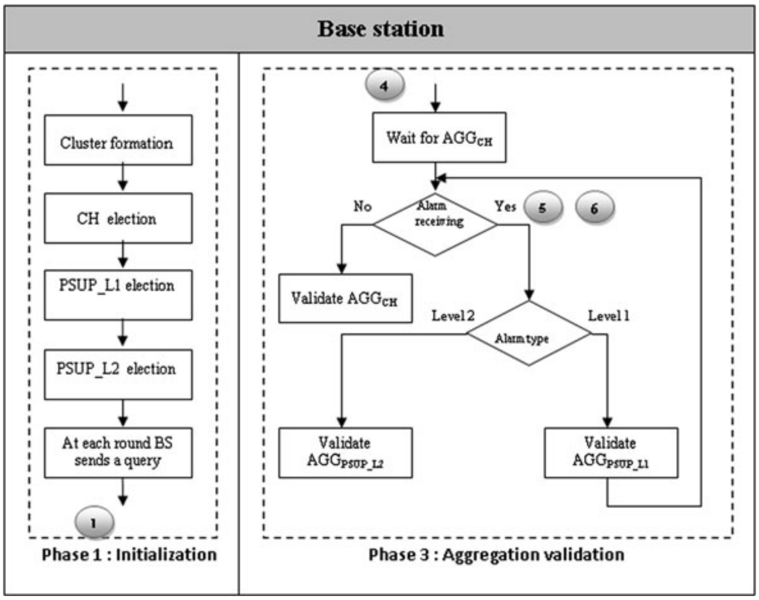

Fig. 1 The flowchart of RAMA: phases 1 and 3

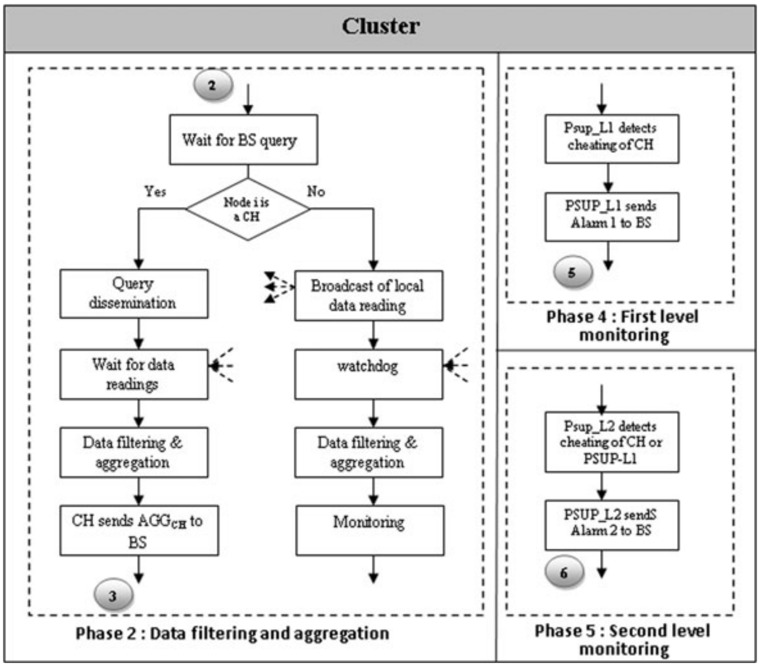
Cluster
2
Wait for BS query
Yes
Node i is a CH
No
Query dissemination
Broadcast of local data reading
Wait for data readings
watchdog
Data filtering & aggregation
Data filtering & aggregation
CH sends AGG$_{CH}$ to BS
Monitoring
3
Phase 2 : Data filtering and aggregation
Psup_L1 detects cheating of CH
PSUP_L1 sends Alarm 1 to BS
5
Phase 4 : First level monitoring
Psup_L2 detects cheating of CH or PSUP-L1
PSUP_L2 sendS Alarm 2 to BS
6
Phase 5 : Second level monitoring

Fig. 2 The flowchart of RAMA: phases 2, 4 and 5

of aggregation result and react only when this accuracy is violated. We assume that the *CH* does not have data itself.

As well as all aggregation protocols available in the literature, aggregation process is done in rounds (synchronization is required). The l_{th} aggregation round on a cluster Cl_{CHi}, headed by cluster-head CH_i, is done as follows:

Each node $i \in Cl_{CHi}$, except CH_i, broadcasts its reading S_i. Note that an attacker cannot 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 mechanism such as CRC (Cyclic Redundancy Check) [13].

$$i \to * : Id_i, S_i \tag{1}$$

Each node $x \in Cl_{CHi}$, receives (collects) all the broadcasted messages, sent by the 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) 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. RAMA uses the median which is statistically robust to outliers [26]. 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 [27]. 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 taken as twice the maximum measurement error [27].

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*. If there exists outliers, the *CH* includes their *Id* in the message sent to the *BS*.

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

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 monitors agree on the AGG_{PSUP_L1}. Then, it 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, $AGG = f(AGG_i | \forall i, Cl_{CH_i})$.

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

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

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 with them to generate and raises 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 $MONIT_i$ that does not agree on aggregate result, and performs a majority vote to generate an alert message.

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

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 length, 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*:

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

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 of $H(AGG_{PSUP\_L2})$ is different from $H(AGG_{maj})$, *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 Sect. 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 RAMA 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 [28]. It is very difficult to detect such attack. However, most of the existing solutions to secure data aggregation assume that the sensor nodes are reporting data truthfully [15] or accept only data reading that is bounded between minimum and maximum values, according to the application [16]. Other protocols, which rely on concept of trust, have emerged recently. Nevertheless, these approaches generate an extra transmission overhead by the periodic exchange of reputation values among 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 MEDIAN, we test it in a simulation environment using Matlab. We consider the scenario of typical temperature-collection application: A group of sensors such as Micas are 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 is generated by the *randn* function. The Peak Attacker is simulated by a function which replaces those sample elements to a common value that corresponds to the proportion determined by k. This replacement is done in the wide surroundings of the real expected value of the sample. To obtain the maximum distortion reachable by the Peak Attacker, we make 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 Fig. 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, 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 behaviour 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 attack**: 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* cannot forge the legal MAC to generate a majority vote, and thus it cannot 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 cannot prevent a bogus data from infecting the global aggregation, leading in cancellation of all steps of aggregation process. Our scheme RAMA 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 [17–19]. Since all the nodes in 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

Restricted to the limited battery power of nodes, energy conservation becomes a critical design issue in wireless sensor networks [38]. The rationale to use RAMA is to conserve energy by requiring no cryptographic operations and no overhead transmission when sensor nodes behave correctly. This rationale is legitimate only if RAMA does not incur much larger energy cost of data transmission than other aggregation protocols, and if energy cost of monitoring with RAMA 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 RAMA.

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 [19] and [26]. 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 [29], TEEN [30] and APTEEN [31], 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 unsecure aggregation method (TAG [4]), 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 unsecure aggregation, our mechanism has only an overhead of four 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 [17], 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 [32]

Table 2 Transmission overhead comparison with 40 % of compromised nodes

Leaf nodes		16	32	64	128
TAG [4]		4.3 KB	8.4 KB	16.6 KB	33 KB
Hu and Evans [17]		10.8 KB	38.4 KB	49.4 KB	159.8 KB
Our scheme	Scenario 1	4.3 KB	8.4 KB	16.6 KB	33 KB
	Scenario 2	4.6 KB	8.7 KB	16.9 KB	33.3 KB
	Scenario 3	8.4 KB	12.5 KB	24.1 KB	47.1 KB
	Scenario 4	8.1 KB	12.2 KB	23.8 KB	46.8 KB

(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 the honest nodes is $t = 10$ (40 % of compromised node). In summary, through analysis and comparison, as we show 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.

7.2 Computation overhead

The most prevalent concern in wireless sensor networks is the limited lifetime [37]. So, ccryptography causes considerable extra consumption of energy, mainly due to packet overhead, which leads consequently to a shorter network lifetime [32, 33]. 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 μJ for calculating a 29-byte packet MAC [34]. However, most of the 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 [19] and [21] induce a high transmission and computation overhead neglecting the 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 the 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* computes 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

Energy is a scarce resource in wireless networks [36]. Overhearing is often considered a cause of energy wastage [35]. 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 among 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 [13]. On one 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 [35], 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 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 except the last recent work [22]. Table 3 also indicates if the protocol is resilient against malicious aggregators and aggregators' failures in columns 3 and 4 respectively. The column 5 indicates the resilience against data rejection the main drawback of almost all existing solutions that focus on integrity of data aggregation. The

Table 3 Data aggregation protocols: comparing the security features

	Aggregation type	Insider attack resilience	Malicious aggregator resilience	Aggregator failure resilience	Data rejection resilience	Defence type	Management policy
SDA [17]	Hop-by-hop	No	Yes	No	No	Proactive	Unique
SIA [18]	Unique aggregator	No	Yes	No	No	Proactive	Unique
WDA [19]	Hop-by-hop	No	Yes	No	No	Proactive	Unique
SDAP [20]	Hop-by-hop	No	Yes	No	No	Proactive	Unique
FAIR [21]	Hop-by-hop	No	Yes	Yes	Yes	Proactive	Unique
RSAP [22]	Hop-by-hop	Yes	Yes	Yes	Yes	Reactive	Unique
RAMA Our solution	CH	Yes	Yes	Yes	Yes	Reactive	Adaptive

column 6 indicates the type of defense. The *reactive defense* is very efficient because the security measurements must be used only when attack existence. The last column denotes the management policy of protocols. By *unique* rule, we refer to the systematic use of cryptographic primitives and the same reaction of the protocol 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. We can clearly demonstrate that our scheme RAMA outperforms the other proposals.

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 NS2 simulator.

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 data packet.

The simulation was run using different scenarios of attacks and 40 % of compromised nodes are inserted in the cluster. 10 queries are initiated by the base station. The simulation results were obtained by calculating the average of all runs.

Table 4 Simulation parameters

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	100 m × 100 m
Transmission range	40 m

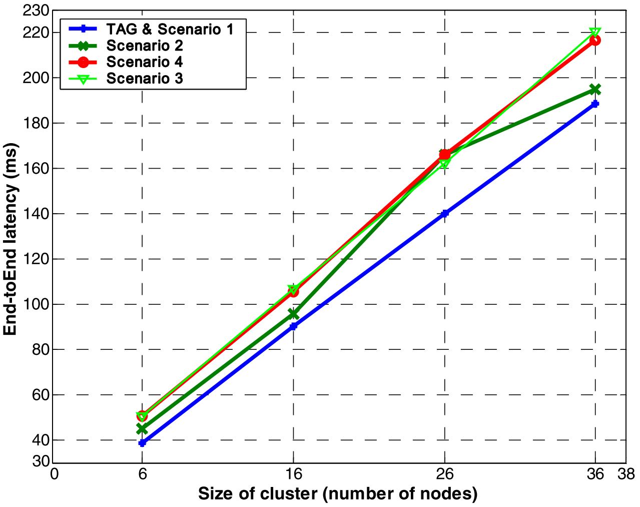

Fig. 3 Latency delivery

For comparison purpose, we also implement the unsecure aggregation protocol (TAG [4]) and classical secure aggregation scheme in which integrity violation induces 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 re-

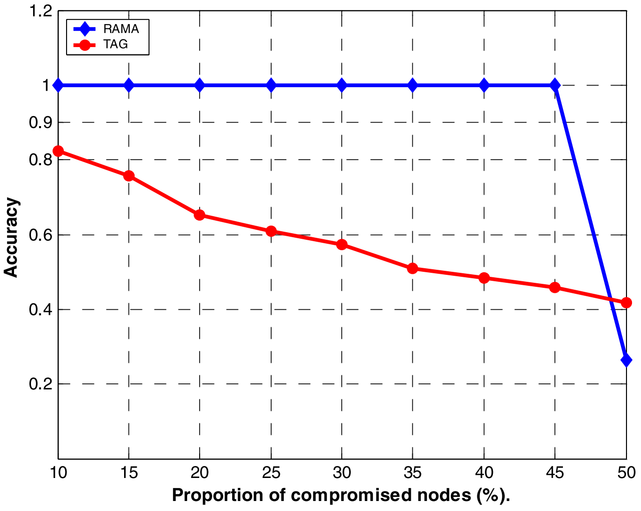

Fig. 4 Accuracy comparison of TAG and RAMA

sult to *BS* without referring to cancel aggregation process when cheating is detected. Comparing with unsecure aggregation (TAG), the delivery speed in our scheme is *constant* and *very close* to TAG in both scenario 1 and scenario 2. However, in scenario 3 and scenario 4, 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, RAMA 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 RAMA 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 unsecure aggregation scheme TAG which is very sensitive in untrusted environment. In RAMA, the accuracy, is very high below of 50 % of compromised nodes. Thus, RAMA (in all attack scenarios) has better accuracy than TAG.

3. Energy efficiency: RAMA 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 RAMA compared to classical secure aggregation scheme.

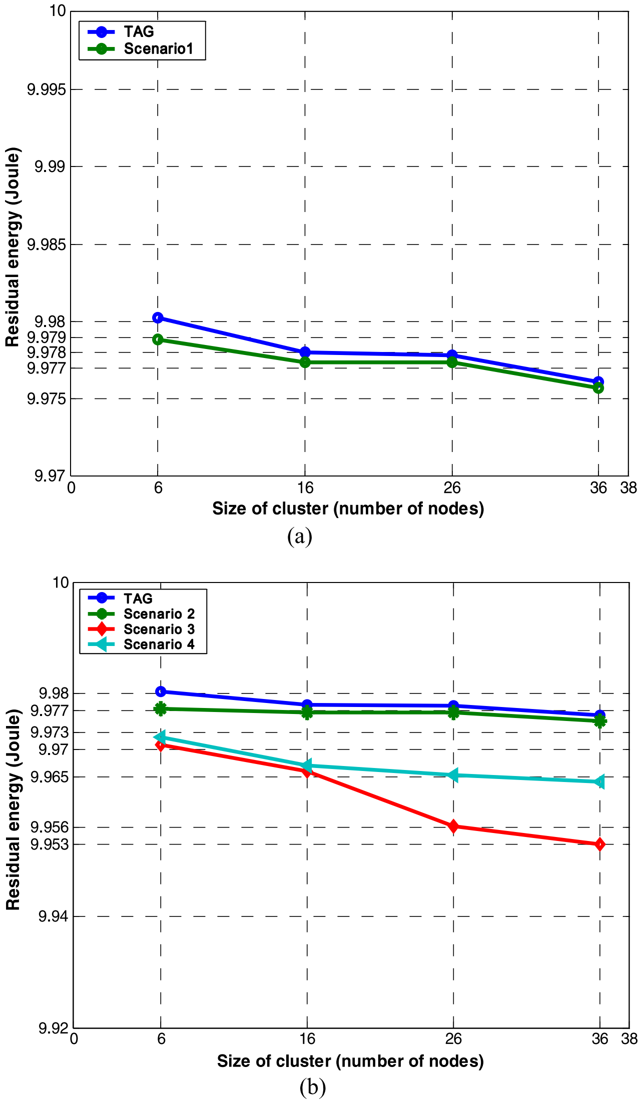

Fig. 5 (**a**) Residual energy in normal situation. (**b**) Residual energy in presence of attack

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. Figures 5(a) and (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 Fig. 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

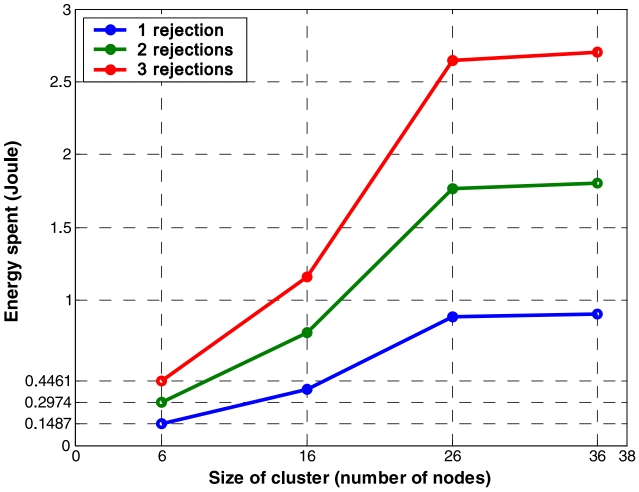

Fig. 6 Energy spent with data rejection

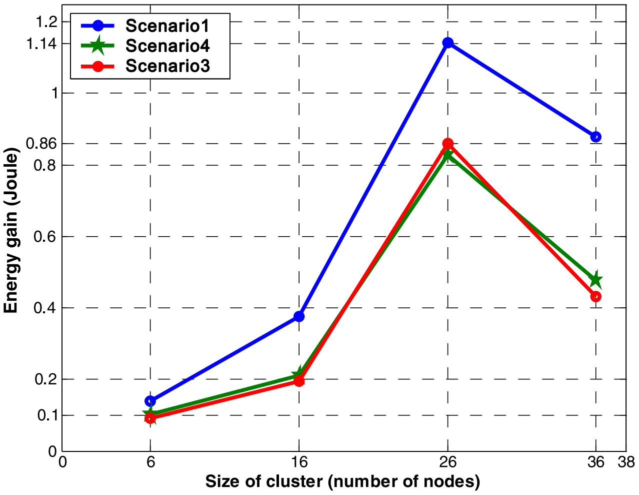

Fig. 7 Energy gain of RAMA (1 rejection)

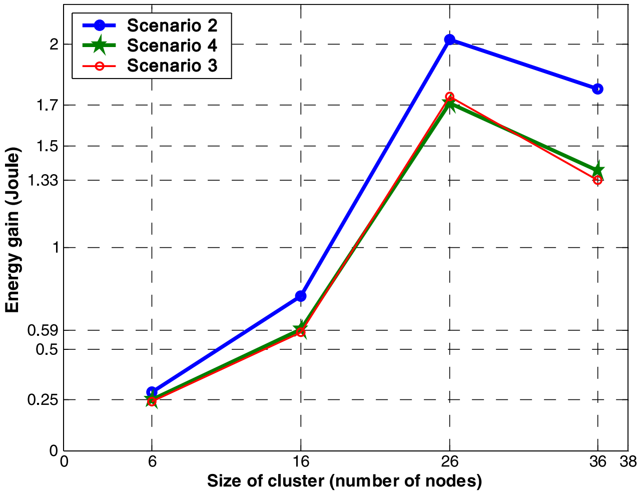

Fig. 8 Energy gain of RAMA (2 rejections)

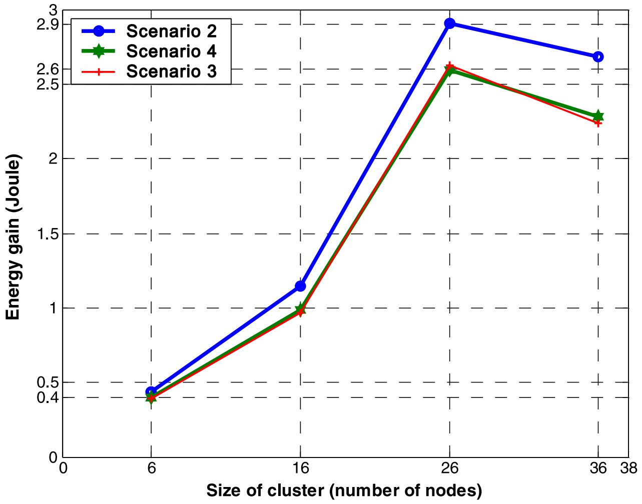

Fig. 9 Energy gain of RAMA (3 rejections)

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, Figs. 7, 8 and 9, illustrate the energy saving by RAMA compared to classical scheme respectively with one, two and three data rejections.

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

8 Conclusion

We have presented RAMA 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, RAMA is robust against bogus data injection and total data rejection and has the ability to recover from aggregator failure without neglecting energy

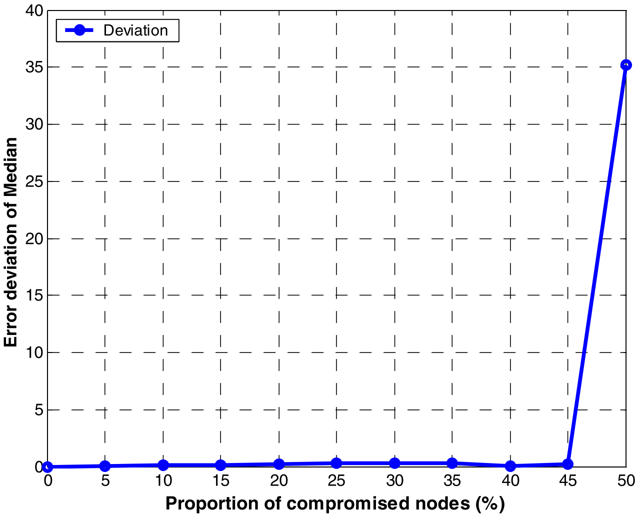

Fig. 10 Error deviation of Median vs proportion of compromised nodes in a cluster

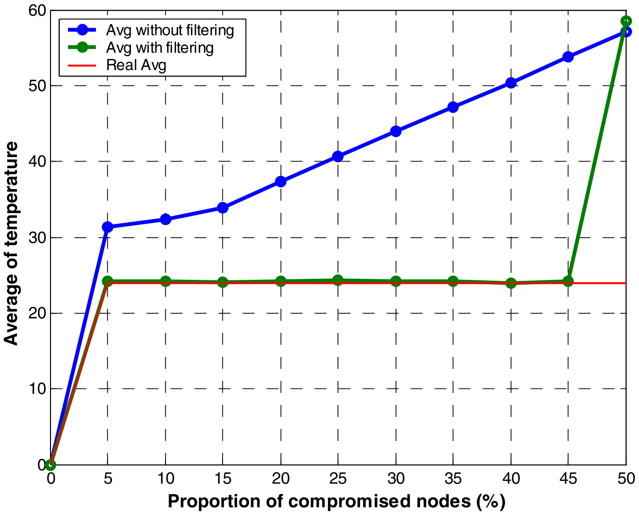

Fig. 11 Comparison of average aggregation vs proportion of compromised nodes in a cluster

efficient, providing thus much higher availability than other security protocols.

References

1. Mármol, F. G., & Pérez, G. M. (2011). Providing trust in wireless sensor networks using a bio-inspired technique. *Telecommunications Systems*, *46*(2), 163–180.
2. Ye, Z., Abouzeid, A. A., & Jing Ai, J. (2007). Optimal policies for distributed data aggregation in wireless sensor networks. In *26th IEEE international conference on computer communications*, May 2007 (pp. 1676–1684).
3. Krishnamachari, B., Estrin, D., & Wicker, S. (2002). The impact of data aggregation in wireless sensor networks. In *22nd international conference on distributed computing systems* (pp. 575–578).
4. Madden, S., Franklin, M. J., & Hellerstein, J. M. (2002). TAG: a tiny aggregation service for ad-hoc sensor networks. In *5th symposium on operating systems design and implementation*, December 2002 (pp. 131–146).
5. Itanagonwiwat, C., Govindan, R., & Estrin, D. (2000). *Directed diffusion: a scalable and robust communication paradigm for sensor networks* (Technical report 00-732). University of Southern California.
6. Deshpande, A., Nath, S., Gibbons, P. B., & Seshan, S. (2003). Cache-and-query for wide area sensor databases. In *2003 ACM SIGMOD international conference on management of data* (pp. 503–514).
7. Solis, I., & Obraczka, K. (2004). The impact of timing in data aggregation for sensor networks. In *IEEE international conference on communications*, June 2004 (pp. 3640–3646).
8. Tang, X., & Xu, J. (2006). Extending network lifetime for precision constrained data aggregation in wireless sensor networks. In *25th IEEE international conference on computer communications*, April 2006 (pp. 1–6).
9. Fasolo, E., Rossi, M., Widmer, J., & Zorzi, M. (2007). In-network aggregation techniques for wireless sensor networks: a survey. *IEEE Wireless Communications*, *14*(2), 70–87.
10. Rajagopalan, R., & Varshney, P. K. (2006). Data-aggregation techniques in sensor networks: a survey. *IEEE Communications Surveys and Tutorials*, *8*(4), 48–63.
11. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, *40*(8), 102–114.
12. Ganeriwal, S., & Srivastava, M. (2004). Reputation-based framework for high integrity sensor networks. In *2nd ACM workshop on security of ad hoc and sensor networks*, October 2004 (pp. 66–77).
13. Ning, P., & Sun, K. (2005). How to misuse AODV: a case study of insider attacks against mobile ad-hoc routing protocols. *Ad Hoc Networks*, *3*(6), 795–819.
14. Perrig, A., Stankovic, J., & Wagner, D. (2004). Security in wireless sensor networks. In *Communication of the ACM*.
15. Labraoui, N., Guerroui, M., Aliouat, M., & Zia, T. (2011). Data aggregation security challenge in wireless sensor networks: a survey. *Ad hoc & Sensor Networks, International Journal*, *12*(3–4), 295–324.
16. Bagaa, M., Lasla, N., Ouadjaout, A., & Challal, Y. (2007). SEDAN: secure and efficient protocol for data aggregation in wireless sensor networks. In *32nd IEEE conference on local computer networks*, October 2007 (pp. 1053–1060).
17. Hu, L., & Evans, D. (2003). Secure aggregation for wireless networks. In *Symposium on applications and the internet workshops*, January 2003 (pp. 384–391).
18. Przydatek, B., Song, D., & Perrig, A. (2003). SIA: secure information aggregation in sensor networks. In *1st ACM international conference on embedded networked sensor systems* (pp. 255–265).
19. Du, W., Deng, J., Han, Y. S., & Varshney, P. (2003). A witness-based approach for data fusion assurance in wireless sensor networks. In *Global telecommunication conference* (Vol. 3, pp. 1435–1439).
20. Yang, Y., Wang, X., Zhu, S., & Cao, G. (2006). SDAP: a secure hop-by-hop data aggregation protocol for sensor networks. In *7th ACM international symposium on mobile ad hoc networking and computing* (pp. 356–367).
21. Cristofaro, E., Bohli, M. J., & Westhoff, D. (2009). FAIR: fuzzy-based aggregation providing in-network resilience for real-time wireless sensor networks. In *2nd ACM conference on wireless network security* (pp. 253–260).
22. Jaydip, S. (2011). A robust and secure aggregation protocol for wireless sensor networks. In *6th international symposium on electronic design, test and applications*, January 2011 (pp. 222–227).

23. Boulis, A., Ganeriwal, S., & Srivastava, M. B. (2003). Aggregation in sensor networks: an energy-accuracy trade-off. *Ad Hoc Networks*, *1*(2), 317–331.
24. Sun, K., Peng, P., Ning, P., & Wang, C. (2006). Secure distributed cluster formation in wireless sensor networks. In *22nd annual computer security applications conference*, December 2006 (pp. 131–140).
25. Ergen, S. C., & Varaiya, P. (2010). TDMA scheduling algorithms for wireless sensor networks. *Wireless Networks*, *4*, 985–997.
26. Wagner, D. (2004). Resilient aggregation in sensor networks. In *2nd ACM workshop on security of ad hoc and sensor networks* (pp. 78–87).
27. Kumar, A., & Ribeiro, V. J. (2009). REEF: a reliable and energy efficient framework for wireless sensor networks. In *1st international communication systems and networks and workshops*, January 2009 (pp. 1–9).
28. Can, H., Perrig, A., & Song, D. (2006). Secure hierarchical in-network aggregation in sensor networks. In *13th ACM conference on computer and communications security* (pp. 278–287).
29. Handy, M., Haase, M., & Timmermann, D. (2002). Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In *4th international workshop on mobile and wireless communications networks* (p. 368).
30. Manjeshwar, A., & Agrawal, D. (2002). TEEN: a protocol for enhanced efficiency in WSN. In *IEEE international symposium on parallel and distributed processing* (pp. 2009–2015).
31. Manjeshwar, A., & Agrawal, D. (2002). APTEEN: a hybrid protocol for efficient routing and a comprehensive information retrieval in WSN. In *IEEE international symposium on parallel and distributed processing* (pp. 195–202).
32. Perrig, A., Szewczyk, R., Wen, V., Cullar, D., & Tygar, J.D. (2001). Spins: security protocols for sensor networks. In *7th annual ACM/IEEE international conference on mobile computing and networking* (pp. 189–199).
33. Karlof, C., Sastry, N., & Wagner, D. (2004). TinySec: a link layer security architecture for wireless sensor networks. In *2nd international conference on embedded networked sensor systems* (pp. 162–175).
34. Wander, A. S., Gura, N., Eberle, H., Gupta, V., & Shantz, S. (2005). Energy analysis of public-key cryptography on small wireless devices. In *3rd IEEE international conference on pervasive computing*, March 2005.
35. Wu, K., Dreef, D., Sun, B., & Xiao, Y. (2006). Secure data aggregation without persistent cryptographic operations in wireless sensor networks. In *21st IEEE international conference on performance computing and communications* (p. 85).
36. Li, H., Chan, E., & Chen, G. (2010). AEETC—adaptive energy-efficient timing control in wireless networks with network coding. *Telecommunications Systems*, *45*(4), 289–301.
37. Dargie, W., Xiaojuan, C., & Denko, M. K. (2010). Modelling the energy cost of a fully operational wireless sensor network. *Telecommunications Systems*, *44*(1–2), 3–15.
38. Cheng, S. T., & Wu, M. (2009). Optimization of multilevel power adjustment in wireless sensor networks. *Telecommunications Systems*, *42*(1–2), 109–121.

Nabila Labraoui is an associate professor in computer engineering at the university of Tlemcen. She received her Ph.D. in computer engineering from the University of Tlemcen. Her current research interests include wireless ad hoc sensor networks, Network Security, Localization and Trust management for distributed and mobile systems.

Mourad Gueroui is an associate professor in computer engineering at the University of Versailles, France. His research interests are in the area of wireless networks (WATM, WLAN and sensors), particularly performance evaluation and QoS provisioning. He received his M.Sc. degree from University of Paris6 and his Ph.D. in networking and computer engineering from the University of Versailles.

Makhlouf Aliouat is an associate professor in computer engineering at the University of Setif, Algeria. His research interests are in the area of wireless networks, particularly security issues and fault tolerance.

Jonathan Petit is a Senior Researcher in the Distributed and Embedded Security group at the University of Twente, Netherlands. He received his Ph.D. degree in Networks, Systems and Architecture from the University of Toulouse, France, in 2011. He is technical coordinator of the European FP7 PRESERVE project. His research interests include security & privacy, ITS, wireless and vehicular.