DARE: evaluating Data Accurancy using node REputation

Sabrina Sicari[a,*], Alberto Coen-Porisini[a,**], Roberto Riggio[b,]

[a] *Università degli studi dell'Insubria*
Dipartimento di Scienze Teoriche e Applicate
Via Mazzini, 5
21100 Varese, Italy
[b] *CREATE-NET*
Via Alla Cascata 56/C, 38123, Trento, Italy

Abstract

Typical wireless sensor networks (WSNs) applications are characterized by a certain number of different requirements such as: data accuracy, localization, reputation, security, and confidentiality. Moreover, being often battery powered, WSNs face the challenge of ensuring privacy and security despite power consumption limitations. When the application scenario allows their use, data aggregation techniques can significantly reduce the amount of data exchanged over the wireless link at the price of an increased computational complexity and the potential exposition to data integrity risks in the presence of malicious nodes. In this paper, we propose DARE, an hybrid architecture combining WSNs with the wireless mesh networking paradigm in order to provide secure data aggregation and node reputation in WSNs. Finally, the use of a secure verifiable multilateration technique allows the network to retain the trustworthiness of aggregated data even in the presence of malicious node. Extensive performance evaluations carried out using simulations as well as a real–world prototype implementation, show that DARE can effectively reduce the amount of data exchanged over the wireless medium delivering up to 50% battery lifetime improvement to the wireless sensors.

Keywords: sensors networks, mesh architecture, secure aggregation, localization, verifiable multilateration, simulations, testbed

[*] sabrina.sicari@uninsubria.it
[**] alberto.coenporisini@uninsubria.it
roberto.riggio@create-net.org

1. Introduction

In the recent years the number of applications using Wireless Sensor Networks (WSNs) has dramatically increased and therefore requirements such as data accuracy, localization, reputation, security, and confidentiality are becoming more and more important in many application scenarios. Moreover, since WSNs are very often battery powered, optimizing the power consumption of wireless sensors nodes is considered of vital importance by both researchers and practitioners.

Thus, it is necessary to design WSNs meeting the above requirements while satisfying the power constraints imposed by the technology. Notice that such requirements are very often related one another. For instance, in a WSN monitoring physical quantities (e.g., temperature), data accuracy depends on nodes position since it is necessary to know where sensors are located in order to have an accurate picture of the status of the monitored environment. Since the position of sensor nodes is often computed by means of nodes cooperation, attacks such as node displacement; wormholes fabricated communication links; distance enlargement by introducing fake nodes; dissemination of false position and distance information by compromising nodes, may lead to an incorrect information about nodes position, threatening in this way the security of the whole WSN.

Typical security and privacy techniques used in wireless networks are not directly applicable to WSNs due to their needs in terms of power consumption. A possible solution is represented by a technique known as Verifiable Multilateration (VM) [1] that allows one to determine the level of trustworthiness associated with the position reported by a sensor node based on the previous behavior of the node itself. In other words nodes are associated with a level of reputation representing trustworthiness. Therefore, data coming from nodes having a good reputation, are considered trustworthy, while data coming from nodes having a bad reputation are not.

Another way to reduce the overall power consumption of a WSN is based on minimizing the number of data transmissions, since this operation is the most energy demanding one. This can be done by using data aggregation techniques [2, 3], which can significantly reduce the amount of data exchanged over the wireless link, while increasing the amount of computation performed by sensor nodes. However, data aggregation raises several privacy and security issues since it is potentially vulnerable to attackers who, for instance, may inject bogus information without being detected. Secure aggregation techniques, such as the one defined by Castelluccia et al [4], which guarantees end-to-end confidentiality and integrity to the aggregated data, can be used to overcome such issues.

However, data aggregation, verifiable multilateration along with other known techniques may not be enough to ensure the level of security required within the power constraints imposed by the WSN technology. In fact, the limited WSN resources in term of power on one hand and the application requirements on the

other hand, call for new solutions. Thus, we decided to move from the traditional architecture comprising only wireless sensor nodes and use a hybrid architecture, that is the combination of two or more network architectures, in order to exploit the capabilities offered by the integration of different technologies.

In this paper, we introduce DARE (evaluating Data Accuracy using node REputation), an hybrid architecture combining WSNs and wireless mesh networks (WMNs) exploiting the gateway/bridge functionalities of mesh routers that allows the integration of WMNs with other networks [5]. Sensor nodes provide only sensing functionalities and they forward sensed data to the closest mesh router. Mesh routers, in turn, provide secure data aggregation and node localization capabilities and are in charge of relaying the aggregated data to the *Sink*. Such an architecture reduces the amount of data exchanged over the network, by splitting the required functionalities between sensor nodes and mesh routers leaving the latter in charge of the more computationally intensive tasks.

We already investigated secure and energy efficient WSNs in some of our previous works [6, 7, 8, 9, 10]. More specifically, in [11] we defined a hybrid solution between wireless sensors and mesh networks to perform secure data aggregation without taking into account node localization nor node reputation. DyDAP [6] presented an approach coupling a privacy management policy with an original aggregation algorithm able to deal with end-to-end encrypted data, without taking into account node localization nor node reputation. In [10] an analysis of malicious node behavior during localization is investigated in depth exploting game of theory concepts, but power consumption and data integrity are not addressed. Thus, DARE extends the results obtained in our previous works defining a hybrid architecture that, in addition to implement secure data aggregation, allows nodes to be localized by using their reputation.

Moreover, an evaluation of the power consumption of DARE architecture and the related battery lifetime is conducted using the energy consumption models presented in [12] . Finally, DARE performances are investigated by means of simulations whose results show that our approach outperforms other solutions. In addition, we developed a prototype to test the practical viability of our approach in realistic settings.

The remaining of the paper is organized as follow: Section 2 summarizes the security model used in this work, while Section 3 describes the DARE network architecture. Section 4 presents the results of the simulation tests, while the results obtained by exploiting a real-world prototype are reported in Section 5. Finally, a brief overview of the state of the art is presented in Section 6; while Section 7 draws some conclusions and provides hints for future works.

2. Security Model

The application domains of WSN are really wide spreading from telemedicine to military applications, from ambient monitoring to smart city applications and so on. A lot of such applications provide services that use average data, for example the average temperature, the average pressure. For such a kind of applications it is possible to reduce the amount of data transmitted over the wireless medium using in–network aggregation techniques. Notice that in this case the power consumption of sensor nodes is reduced because sensor nodes use more power during the transmission and reception communication phases than when performing computation [13]. Thus, aggregation protocols may help in reducing the overall traffic among nodes. At the same time, since nodes are the attack goals of malicious users who try to violate the confidentiality and the integrity of data, proper countermeasures are needed to perform a secure data aggregation. Encryption can be used to secure node communication, both hop-by-hop and end-to-end secure data aggregation are supported. In the former case, the data are encrypted by sensing nodes and decrypted by aggregators. The aggregator nodes, then, decrypt the data coming from the sensing nodes, aggregate them, and encrypt them again, until eventually the *Sink* node gets the final encrypted aggregation result (and decrypts it). In the end-to-end approach the intermediate aggregators manipulate only encrypted data and they have no keys to decrypt them. In our work we consider applications that use the aggregated data based on an operation of sum of sensing data. For this kind of data it is possible to use, for example, the additively homomorphic aggregation model define by Castelluccia et al. [4], which allows encrypted data to be aggregated without decrypting the data hop–by–hop. We chose this end-to-end secure aggregation solution in which an attack to any aggregator node is not able to compromise the whole system.

Beside reducing traffic amount in secure manner there is another requirement related to the node positition, which requires to be computed by node cooperation (i.e., nodes exchange information in order to allow an estimation). The node positions can be evaluated by using a multilateration technique, which determines the node coordinates by exploiting a set of landmark nodes, called the *anchor nodes*, whose positions are known. However, the node position estimation should be object of different kind of security attacks. In order to address such security problems in literature many solutions are available. In DARE we adopt the Verifiable Multilateration (VM) [1] for its capability in classifying malicious node behavior. Notice that the robustness of VM is investigated in depth in our previous work [10].

Summarizing, our security model is composed of two main blocks: the end–to–end secure data aggregation scheme exploiting additively homorphic encryption [4] and the verifiable multilateration [1] technique. The former guarantees the confidentiality and integrity of the aggregated data, while the latter allows

us to identify malicious nodes. In this section we briefly describe the end-to-end secure data aggregation algorithm and the VM algorithm.

2.1. Secure aggregation

The algorithm of Castelluccia et al. [4] is based on a simple and secure additively homomorphic stream cipher that allows efficient aggregation of encrypted data. Homomorphic encryption schemes are especially useful in scenarios where someone, having no decryption keys, needs to perform arithmetic operations on a set of ciphertexts.

The cipher uses modular additions and is therefore very well suited for CPU-constrained devices like sensors. Moreover, aggregation based on this cipher can be used to efficiently compute statistical values such as mean, variance, and standard deviation of sensed data, enabling significant bandwidth gain.

The main idea of [4] is to replace the XOR (Exclusive-OR) operation, typically found in stream ciphers, with modular addition. For reader convenience, we will briefly sketch the additively homomorphic encryption scheme proposed in [4] by applying it in the context of WSN.

Let us consider a network comprising N nodes, each of which is uniquely identified by a label n_i, $1 \leq i \leq N$. Moreover let X_i denote an integer number representing the data measured/sensed by node n_i, where $X_i \in [0; M-1]$and M is a large enough integer, whose value is discussed later on.

Thus, the encrypted ciphertext c_i of datum X_i measured by node n_i is given by

$$c_i = Enc(X_i, k_i) = X_i + k_i \ (mod(M)). \tag{1}$$

where *Enc*, *mod* and k_i respectively represent the encryption operation, the modulus operation and the node key used for encryption/ decryption operation.

The aggregation of J different ciphertexts $c_1, ..., c_J$ received from other nodes is carried out in the following way:

$$c_{aggr} = \sum_{i=1}^{J} c_i \ (mod(M)) \tag{2}$$

Since the above encryption scheme is additively homomorphic, we have that if $c_1 = Enc(X_1, k_1)$ and $c_2 = Enc(X_2, k_2)$ then $c_1 + c_2 = Enc(X_1 + X_2, k_1 + k_2)$.

As a consequence, the cleartext of the aggregated data X can then be obtained by:

$$X = Dec(c_{aggr}, k) = c_{aggr} - k \ (mod(M)); \qquad k = \sum_{i=1}^{J} k_i \ . \tag{3}$$

where *Dec* represents the decryption operation, while K is equal to the sum of the keys of the nodes whose sensed encrypted data are aggregated.

In order to prove its viability in a realistic scenario, we implemented a specific use case on top of our hybrid architecture. The ensuing application computes the average and the variance of the physical phenomena monitored by the WSN (e.g., the temperature).

In the application scenario envisioned in this work, each sensor node periodically samples the environmental temperature. The collected data is then forwarded to the aggregator node through the sensors, where the secure aggregation scheme is implemented. In order to obtain average and variance, sensor nodes are required to compute:

$$S = \sum_{i=1}^{n} X_i \quad V = \sum_{i=1}^{n} X_i^2 \tag{4}$$

where X_i is the individual value measured by a sensor node and n is the total number of answering sensors. The *Sink* will then receive two distinct values, which can be used to compute both the average $E(x)$ and the variance $Var(x)$:

$$E(x) = \frac{\sum_{i=1}^{n} X_i}{n} \quad E(x^2) = \frac{\sum_{i=1}^{n} X_i^2}{n} \tag{5}$$

$$Var(X) = E(x^2) - E(x)^2 \tag{6}$$

It is worth noting that, in computing the average, the modulus M must be large enough to prevent any overflow. The modulus is thus chosen as follows: $M = n * p$, where $p = max(m_i)$ is the maximum value that can be assumed by the message, and n is the total number of sensor nodes in the network. Therefore each ciphertexts will be $log(M) = log(p) + log(n)$ bits long. Moreover, if also the variance of the measured data has to be derived an additional modulus M' is necessary for the sum of the squares. As for the average, also M' must be large enough to prevent overflow and it is then chosen as follows: $M' = n * p^2$. The size of the ciphertext is therefore $log(M') = 2 * log(p) + log(n)$ bits.

Two strings, each of them 32 bits long, have been used to encode, respectively, the sum of the values reported by each sensor node ($\sum_{i=1}^{n} X_i$) and sum of their squares ($\sum_{i=1}^{n} X_i^2$). Setting the maximum number of sensor nodes allowed in the WSNs to $n = 2^8 = 256$, leaves us with 24 bits to represent p^2. As a result, we have the following constraint on the range temperatures that can be represented: $m_i \in [0, 2^{12}]$. In fact, in order to represent the square of the maximum value that can be assumed by m_i ($2^{12} = 4096$) without incurring in any overflow, 24 bits are necessary. Hence, notice that in order to correctly decrypt it is important to provide the identifications *id* of the node involved in the aggregation process. In fact, such an information is fundamental in order to identify the node keys which are used in equation (3), more details are discussed in next Section.

The keystream k can be generated using a streamcipher, such as RC4, keyed with a node secret key and a unique message. Finally, each sensor node shares a unique secret key with the *Sink* of the WSN. Such keys are derived from a master secret (known only to the *Sink*) and distributed to the sensor nodes. However, the key distribution protocol is outside the scope of this work.

2.2. Secure localization

Node positions are derived using a multilateration technique which determines the position of a node by exploiting a set of landmark nodes, called *anchor nodes*, whose absolute positions are known. The position of an unknown node u is computed using an estimation of the distances between the anchor nodes and u. Notice that such distances are computed by measuring the time needed to successfully receive a reply from node u to the beacon messages previsouly broadcasted by each anchor node.

In case node u behaves maliciously, the only way in which it may pretend to be in a location different to the actual one is by delaying the reply to the beacon message. However, under some conditions, it is possible to detect such malicious behaviors by using the Verifiable Multilateration (VM) technique presented in [1], which uses three or more anchor nodes to detect misbehaving nodes. In the rest of this section we briefly summarize the VM operating principles.

Let V_1, V_2, and V_3 be the anchor nodes (i.e., the verifiers) and let be u the node whose position is unknown. Moreover, let us assume that u lies in the triangle formed by V_1; V_2; V_3. If u tries to pretend to be farther away from one anchor then it has to pretend to be closer to another one. In order to achieve this goal, node u would be required to know the relative position of every anchor node in the network. However, since such information is not available to node u, it is possible to detect malicious nodes. More specifically, let be $T1$, $T2$ and $T3$ be the time needed to get an answer from u to the beacon message sent by $V1$, $V2$, and $V3$, respectively. Starting from T_i the corresponding distance is computed, for $1 \leq i \leq 3$. Let x_u, y_u denote the coordinates of the estimated position of u, and let $fi(x_u, y_u)$ denote the function representing the difference between the distance bound and the estimated distance of u from V_i.

$$fi(x_u, y_u) = \epsilon = \sum_i (db_i - \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2})^2 \tag{7}$$

Finally, the estimated position of u is computed using the minimum mean square estimate (MMSE) that is by minimizing:

$$F(x_u, y_u) = \sum f(x_i, y_i) \tag{8}$$

Once computed, the estimated position of u undergoes two different tests before being considered as reliable. The first test, known as δ–test, aims at

verifying whether the estimated position is compatible with the distance bounds previously computed, while the second test, known as point–in–the–triangle–test, aims at verifying whether the estimated position of u lies inside the triangle formed by the three verifiers. More specifically:

- δ–test: Let δ_{err} denote the maximum distance measurement error allowable; therefore the position of$u$, $< x_u, y_u >$, is considered correct if $f_i(x_u, y_u) < \delta_{err}$, for $1 \leq i \leq 3$. If the test fails then at least for one V_i the estimated distance differs from the distance bound by more than allowed error. In a such a case the estimation is considered to be affected by malicious tampering and therefore node u is marked as *Malicious*.

- Point–in–the–triangle–test: Distance bounds can be used in the previous test only if u lies inside the triangle formed by the three verifiers, otherwise the position of u is considered unverified and therefore node u is marked as *Unknown*.

If both tests are passed, the estimated position is considered as correct and therefore node u is marked as Robust. The final reputation of each node belonging to the cluster is stored in a table by the own mesh router.

3. Network Architecture

The network architecture is composed of clusters of sensor nodes that exploit a wireless multi–hop mesh backbone in order to deliver their data to the *Sink*. Clusters are composed of a variable number of sensors, one of which acting as *Sensor Head*, and one mesh router acting as *Cluster Head*, see Figure 1.

Sensor Heads are responsible for aggregating encrypted messages originated from the sensor nodes in their cluster, while *Cluster Heads* implement the secure aggregation scheme also of data coming from different clusters. Notice that multi–hopping can be exploited by sensor nodes in order to establish connectivity with their *Sensor Head*. *Cluster Head* and *Sensor Head* roles can be implemented by two different nodes, one equipped with a WSN interface (e.g.,IEEE802.15.4) and the other equipped with a WMN interface (e.g., IEEE 802.11). In particular deployment scenario *Sensor Head* and *Cluster Head* roles can be collapsed onto the same network element, typically powerfull mesh router.

In fact the presence or absence of *Sensor head* allows or not to perform two different kind of aggregation: *In-Cluster Aggregation* and *Aggregation*. The former requires the presence of a *Sensor Head* and aggregates data coming from sensors belonging to the same cluster. The related message is named IAMEX (In-cluster Aggregated Message) and it is generated by *Sensor Head*. When performing *In–cluster Aggregation*, each sensor concatenates the *ids* of the messages

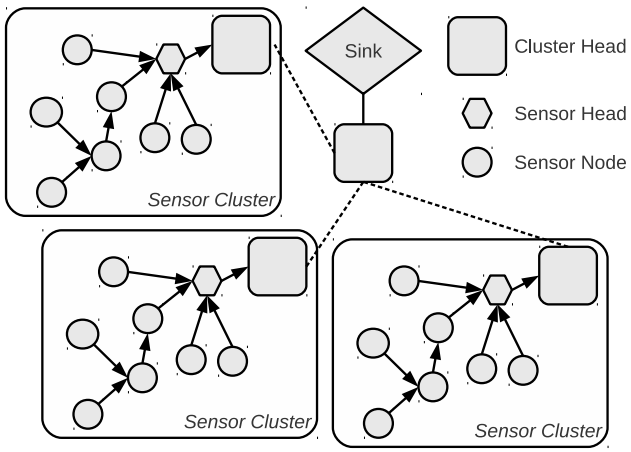


Figure 1: Reference network model for the hybrid mesh/sensor secure aggregation scheme.

being relayed creating in fact a new In-Cluster Aggregated Message (IAMEX). It is worth noticing that, if a locally generated sample is added to the aggregated ciphertext then also the local sensor's *id* will be appended to the IAMEX message in order to satisfy the equation 3. While, in case of *Aggregation* the data, also coming from sensors belonging to different clusters, are aggregated. The related message is named AMEX (*Aggregated Message*) and it is generated by the *Cluster heads*.

The communication is based on a polling schema implemented at the *Cluster Head*. Notice that our design does not require all sensor nodes to reply, on the contrary nodes can fail to reply due to several reasons, e.g., a temporary lack of connectivity, a limited battery, or simply hardware failures or a malicious removal. On the other hand, the network *Sink* must know the *ids* of the non-responding sensor nodes in order to decode the cleartext message, as we just said in previous Section. In order to address such a problem the *Aggregated Message* (AMEX) contains a list of the non–responding nodes in a certain cluster. Such a list can be easily computed by the *Cluster head* using the message received from the sensor nodes and the list of sensor nodes in its cluster (obtained using an initial raging procedure); while the IAMEX message contains the list of the responding nodes.

The *Cluster Head*'s architecture is depicted in Figure 2. Continuous and dashed lines represent communication paths exploiting, respectively, IAMEX and AMEX messages. Notice that, thanks to the homomorphic additively encryption scheme, messages of the same type can be aggregated in a end–to–end fashion by simply adding their ciphertexts and appending the nodes' *ids*.

With regard to the trustworthiness of the nodes, each *Cluster Head* uses a node reputation table reporting, for each sensor node, the trustworthiness of its localization data, information gathered during the localization phase, according

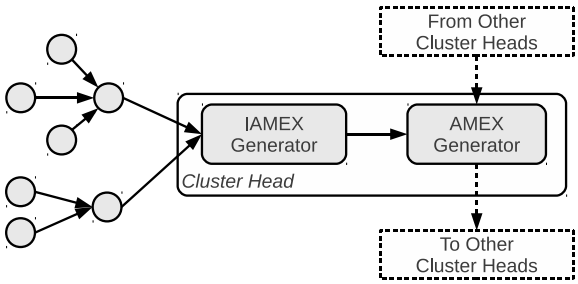


Figure 2: Architecture of the *Cluster head*.

to security model. i.e., Robust, Malicious or Unknown. Notice that initially, anchor nodes (i.e., verifiers) are considered to be *Robust*; while the remaining nodes are classified as *Unknown*. If the reputation is *Robust*, then the *Mesh router* is allowed to treat the data as reliable and aggregate it; if the reputation is *Malicious* the data is discarded; while if the reputation is *Unknown* the data may be processed or discarded depending on the mesh router default policy. Notice that in case of *In–cluster Aggregation* the use of Verifiable Multilateration is not implemented. In fact, in order to implement VM, a *Sensor Head* would be required to receive the reputation table from the relevant mesh routers. However, since the aim of this work is a strict sharing of tasks between sensor nodes and mesh routers with the purpose of possibly improving the sensor nodes' battery lifetime, this use case has not been implemented. In case there is a *Sensor Head* , such a node is directly connected to the *Cluster Head*.
The entire procedure, exploited in order to securely convey and aggregate the samples collected by the sensor nodes to the network *Sink*, can be decomposed into the following steps:

1. The *Cluster Head* periodically polls all the sensors in its cluster. Polling packets can be either flooded across the entire cluster or, if broadcast is not supported, they can be sent using unicast transmissions.
2. Upon polling, each sensor generates a packet containing a single encrypted sample that is then forwarded to the *Cluster Head*.
3. The *Cluster Head* receives the packets coming from sensor nodes in its cluster, evaluate the node repuations according to the node behavior information, stored in the table and if a sensor node is classified as *Robust* based on VM results, it stores the received packets in a local queue, otherwise it discards them. When N packets are received or when a timeout has expired, the *Cluster Head* aggregates its encrypted samples and generates a *AMEX* packet that is sent to the *Sink*.
4. The *Sink* receives all the *AMEX* packets, deciphers the ciphertext and computes the relevant statistical information (e.g. average and/or variance).

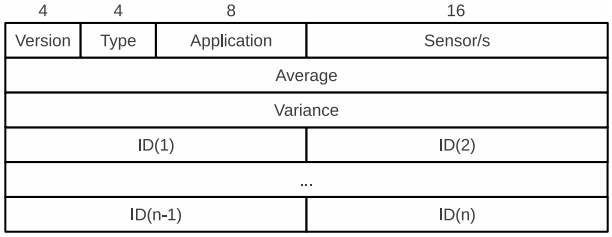


Figure 3: Message format used in our secure aggregation scheme.

3.1. Message Structure

The message structure, used in order to implement the secure aggregation scheme, is illustrated in Figure 3. It consists of 6 fields, plus an optional list of sensor nodes *ids* appended at the end of the message and used only in the AMEX and the IAMEX message types. The fields in the header are packed with the most significant byte first (big endian). Here, follows a detailed description of the different fields:

- *Version (4–bits).* The protocol version (set to 0).
- *Type (4–bits).* The message type:
  - *IAMEX.* Aggregated message emitted by a *Sensor Head*. The *Sensor/s* field contains the number of sensors that contributed to this value. The header is followed by the *ids* of the nodes whose samples have been summed to produce the aggregated value.This kind of message is generated in case an in–cluster aggregation is performed.
  - *AMEX.* Aggregated message emitted by a *Cluster head*. The *Sensor/s* field contains the number of sensors that failed to produce a sample. The header is followed by the *ids* of the non–responding nodes.
  - *Sink.* Sink message emitted by a *Sink*. This message contains the aggregated value in cleartext. The *Sensor/s* field contains the number of sensors that contributed to this value.
- *Application (8–bits).* Used to distinguish among different set of monitored information (e.g. humidity, pressure, etc.). It can be used to map up to 256 different WSN applications over the same mesh–backhaul.
- *Sensor/s (16–bits).* Different meanings according to the particular message type, as described above.
- *Sum (32–bits).* Sum of the readings produced by the sensor node/s.

- *Square (32–bits).* Sum of the squares of the readings produced by the sensor node/s.
- *ID(i).* List of sensor nodes' *ids* (16–bit each). Their meaning depends on the particular message type.

Please note that padding is used in order to ensure that the whole message contains an integral number of 32-bit words, for suppporting, as just explained in details in Section 2, the encoding of average and variance of the values measured by sensors.

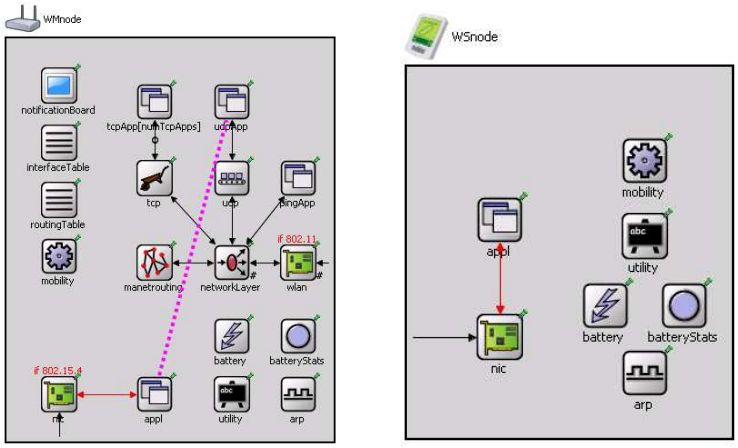
4. Simulations

In this section we report on the outcome of our simulations tests which aimed at assessing DARE's performance with regard to bandwidth efficiency, power consumption, and security. More specifically, we evaluate: the bandwidth efficiency by means the evaluation of the packet number; the power consumption and the related battery lifetime using the energy consumption models presented in [12] and, finally, the level of security that can be obtained by exploiting node infomation maliciousness during the localization phase. The hop–by–hop (HBH) aggregation scheme discussed in [4] is not considered in that, albeit characterized by a slightly higher bandwidth transmission gain, it does not address the end–to–end security concerns.

Notice that, the evaluation of the data confidentiality and integrity features supported by our hybrid architecture has already been provided by Castelluccia et al. in [4] and is thus out of scope for this work. Likewise, the reliability of the verifiable multilateration technique is extensively evaluated in [1] [10] and it is thus also beyond the goals of this work.

4.1. Simulation Environment

The simulations were carried out using the OMNET++ simulator (version 4.1). The INETMANET and the MiXiM models have been used in order to simulate respectively the IEEE802.11–based mesh backhaul and the IEEE802.15.4–based sensor clusters. Each cluster is composed of one mesh router (See Figure 4a) equipped with two radio interfaces and one or more wireless sensor/s (See Figure 4b) each of them equipped with a single radio interface. The primary mesh router interface, derived from the INETMANET framework, is an IEEE 802.11 (WiFi) interface operating in the ISM 2.4 GHz frequency band while the secondary, derived from MiXiM framework, is an IEEE 802.15.4 interface operating in ISM 868 MHz frequency band. It is worth stressing that, the mesh router being equipped with two different interfaces implementing both *Sensor Head* and *Cluster Head* functionalities. Mesh connectivity is implemented by means of the



(a) Mesh router equipped with two interfaces.

(b) Sensor node equipped with a single interface.

Figure 4: Simulation environment's setup.

AODV mesh routing protocol. The sensor nodes are deployed in a star topology around the mesh router. One mesh router acts as a gateway implementing *Sink* functionalities.

4.2. Simulation Scenarios

The following scenarios have been considered:

- *No–Agg.* In this scenario, when a *Cluster Head* receives an encrypted packet from either its sensors cluster or neighbouring cluster head, it immediately forwards it to the *Sink*. No aggregation is performed in this scenario which serves as baseline for the rest of the evaluation.

- *Agg N without VM.* Every packet received by the *Cluster Head* is stored into a FIFO queue. After the N^{th} arrivals the queue is emptied and an *AMEX* packet is generated and forwarded to the *Sink*. The values of N considered for this study have been $4, 8, 12$. The Verifiable Multilateration technique is not used in this scenario.

- *In–cluster Agg N without VM.* Packets forwarded by sensor nodes within a certain cluster are aggregated at each hop in a *IAMEX* message. *IAMEX* messages received by *Cluster Heads* are forwarded to the *Sink*. The Verifiable Multilateration technique is not used in this scenario.

- *Agg N with VM.* Every packet received by the *Cluster Head* is stored into a FIFO queue. Then, each 10 seconds the *Cluster Head* verifies each node'

reputation: messages coming from *Malicious* node are discarded; messages coming from *Unknown* nodes are forwarded towards the *Sink* without being aggregated; and finally messages generated from *Robust* nodes are aggregated and an *AMEX* packet is generated and forwarded to the *Sink*.

Simulations results refer to a network setup consisting of 2 sensor clusters each of them containing one mesh router and 50 sensors distributed over a 500x500 meters square field where mesh routers and sensors nodes are randomly distributed at initialization time. Both mesh and sensor nodes are not mobile. The malicious nodes are randomly distributed and the simulations tests are conducted with different percentage of malicious nodes. Simulation time was set to 300s for all scenarios. The results reported in this work are the average of 10 runs executed with different seed values for the random number generator.

4.3. Energy consumption models and related results

The energy consumption models have been derived using an empirical evaluation of the power consumption of typical wireless devices using *Energino* [12], a real-time energy consumption monitoring toolkit. *Energino's* high performances in term of both sampling frequency and resolution allow to precisely isolate the impact of specific traffic patterns on the overall energy consumption of wireless devices such as Access Points and/or sensor nodes.

For readers' convenience, the empirical power consumption model presented in [12] are briefly sketched here. In the following, $R_{\cdot}(x)$ and $S_{\cdot}(s)$ are the models which accounts for power consumption at the wireless node as function of, respectively, traffic rate and datagram size. Where x is the amount of traffic transmitted or received by the wireless node (expressed in Mb/s) and s is the datagram size (expressed in bytes). Notice that the notation $\cdot = Tx, Rx$ refer to the scenario when the wireless node is acting as transmitter and receiver, respectively. The $R_{\cdot}(x)$ and the $S_{\cdot}(s)$ models are:

$$R_{\cdot}(x) = \begin{cases} \alpha(s) \cdot x + \beta & \text{if } 0 \leq x \leq h(s) \ \text{ Mb/s}, \\ \gamma & \text{if } x > h(s) \ \text{ Mb/s}, \end{cases} \tag{9}$$

$$S_{\cdot}(s) = \begin{cases} -\delta(x) \cdot s + \epsilon(x) & \text{if } p \leq s \leq q \ \text{ byte}, \\ \eta(x) & \text{if } s > q \ \text{ byte}, \end{cases} \tag{10}$$

The parameters have the following physical meaning:

- $\alpha(s)$ $[\mu J/b]$ is the amount or energy spent by the wireless device in order to transmit or receive 1 bit from the session layer with a datagram size of s bytes;

Table 1: R-Model parameters ($s = 1000$ bytes).

	$\alpha(s)$ $[\mu J/b]$	$\beta(s)$ $[W]$	γ $[W]$	$h(s)$ $[Mb/s]$	RMSE $[W]$
$f_{TX}(x)$	0.0259	3.8206	4.6543	32	0.0019
$f_{RX}(x)$	0.0155	3.83	4.2318	26	0.0001

Table 2: S-Model parameters ($x = 10$ Mb/s).

	$\delta(x)$ $[\mu W/b]$	$\eta(x)$ $[W]$	$\epsilon(x)$ $[W]$	q q	RMSE $[W]$
$f_{TX}(s)$	0.0022	4.066	4.900	384	0.0114
$f_{RX}(s)$	0.00079	3.9693	4.4751	640	$3.9165 \cdot 10^{-4}$

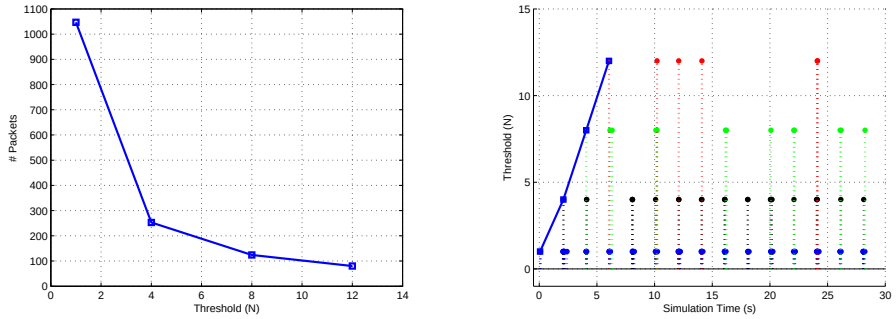
- β $[W]$ is the amount of power consumed by the wireless node in idle mode;
- γ $[W]$ is the maximum amount of power consumed by the wireless node and represents the saturation power consumption;
- $\delta(x)$ $[\mu W/bytes]$ is the amount of power consumed by wireless node in order to transmit or receive 1 byte from the session layer arriving at a rate of x Mb/s;
- $\epsilon(x)$ $[W]$ is the maximum power consumed by the wireless node, transmitting at x Mb/s, using extremely small packets.
- $\eta(x)$ $[W]$ is the minimum power consumed by the wireless node to transmit traffic at a rate of x Mb/s.

Table 1 reports the parameters for R–Model obtained using a datagram size $s = 1000$ bytes, while Table 2 reports the parameters for the new logarithmic S–Model obtained for a transmission rate $x = 10$Mb/s.

4.4. Simulation Results

In this section we discuss the simulations results obtained for our 4 reference scenarios, namely: *No–Agg*, *Agg N without VM*, *In–cluster Agg N without VM*, and *Agg N with VM*.

Figure 5a reports the number of packets delivered to the *Sink* over the WiFi interface during the entire simulation time for increasing values of the aggregation threshold N without using the VM techniques. As expected, increasing the value of N results in a significant reduction in the number of packets delivered to



(a) Number of packets delivered to the *Sink* for increasing values of N.

(b) Inter–arrivals times of packets at the *Sink*.

Figure 5: Outcomes of the simulations campaign.

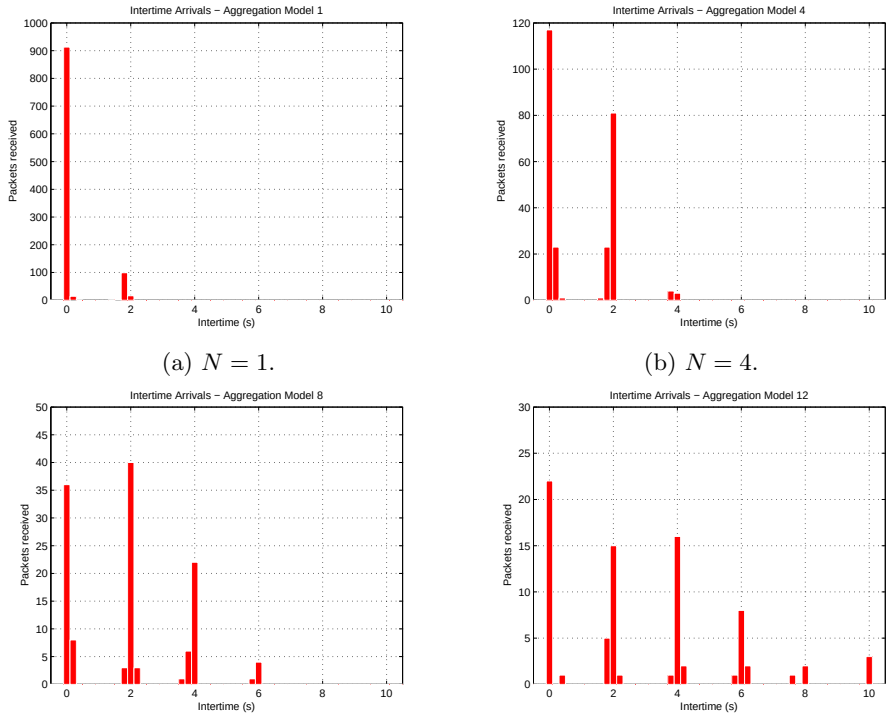
the *Sink*, and thus forwarded across the network which in time results in a lower channel utilization and energy consumption. The inter–arrivals times of aggregated packets are reported in Figure 5b. Notice that, the initial transition time is strictly related with the *Cluster Head*'s polling period, which for this simulations was set to 2 seconds.

Figure 6 reports the histogram of the AMEX messages' inter-arrival times for different values of the aggregation threshold. As it can be noticed, the inter-arrival time increases with the value of N, in particular for $N = 12$, intervals as long as 10 seconds can be observed.

The trustworthiness of the localization data can be enhanced using the VM technique described in Section 2.2. Figure 7 reports the number of message aggregated, forwarded and discarded versus a decreasing number of verifies (70, 50, 30, 10). Results show that, thanks to DARE, data integrity is preserved, in fact, data sent from node with a bad reputation, i.e. *Malicious* node, is discarded; data sent from node uncertain reputation, i.e *Unknown* node, is forwarded to the *Sink* without aggregation and evaluated according to the application domain and ad-hoc policies. Only messages coming from node with a good and verified reputation, i.e. *Robust* node, are aggregated.

As it can be seen from Figure 7 there is a direct correlation between the number of verifiers and the number of messages that can be aggregated. More specifically, an increase of the verifiers number is strictly related to an increase of the number of messages that are suitable for aggregation. On the other hand, lowering the number of verifiers causes more nodes to be marked as *Unknown* leading to an increase in the number of messages forwarded without being aggregated.

Finally, it is also worth noticing that, using a high number of verifiers does not guarantee better performance in terms of number of aggregated messages.



(a) $N = 1$. (b) $N = 4$.

(c) $N = 8$. (d) $N = 12$.

Figure 6: Histogram of the AMEX messages' inter-arrival times for different values of the aggregation threshold N.

For example, from Figure 7 it can be seen that the configuration exploiting 10 verifiers performs better than the configuration exploiting 30 verifiers, since the role performed by the verifiers strictly depends on their position in the network. As a result a small number of well–positioned verifiers can perform better than an higher number of verifiers deployed randomly.

Figure 8 and Figure 9 show the power consumption and battery lifetime respectively, using, as just we said, the model *Energino*.

The energy model is also used to study the power consumption and battery lifetime in the case of an *In–Cluster Agg N without VM* scenario It is important to notice that in this scenario the empirical power consumption model has been applied to the cluster of sensor nodes. Results, reported in Figure 10, refer to a single wireless sensor node. As it can be seen, the proposed aggregation scheme leads to a significant power consumption saving and resulting in extension of a sensor battery lifetime in comparison with a *No–Agg* scenario.

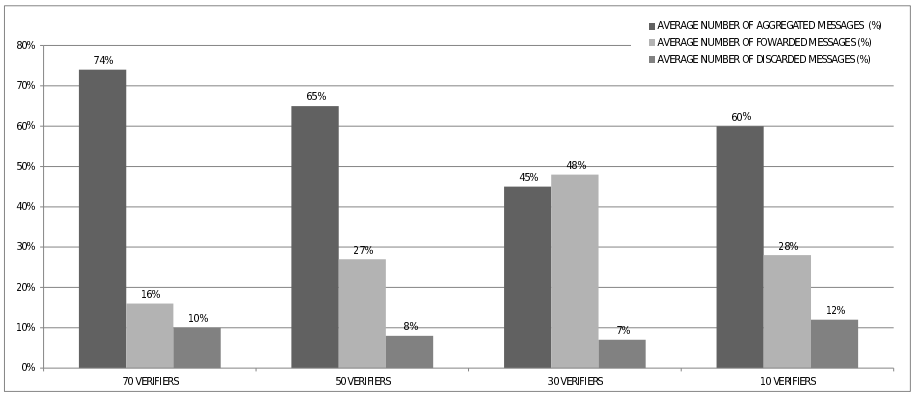


Figure 7: Number of message aggregated, forwarded and discarded versus a decreasing number of verifies.

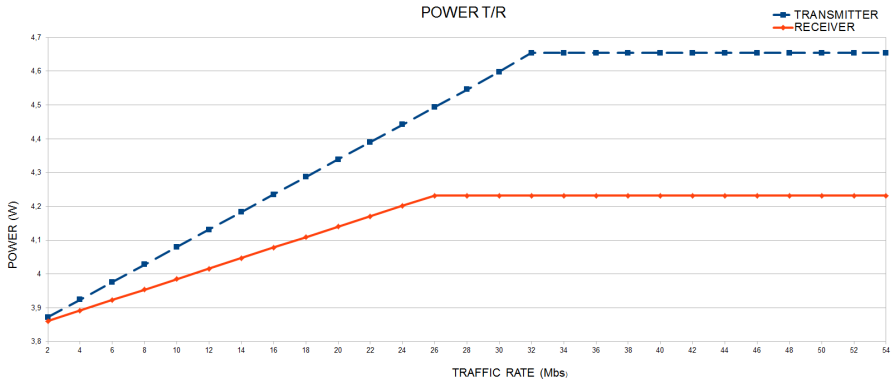


Figure 8: Average power consumption at the wireless sensor node as a function of the bitrate for a constant datagram length of 1280 bytes.

5. Prototype

A prototype has also been implemented and tested in order to demonstrate the practical viability of our approach in realistic settings. This study has been conducted exploiting 4 mesh routers organized in a linear topology (see Figure 11) and implementing both *Cluster Head* and *Sensor Head* functionalities. A Dell D630 laptop, connected through an Ethernet cable to the fourth *Cluster Head*, has been exploited as network *Sink*. Sensor nodes have been emulated by means of a software process running within each mesh router. This process emulates a flat WSN computing both the average and the variance of the physical phenomena monitored by the WSN (e.g., the temperature). Each sensors cluster is composed of 60 nodes. The mesh backhaul has been implemented using the WING toolkit, an experimental IEEE 802.11 wireless mesh network [14]. No

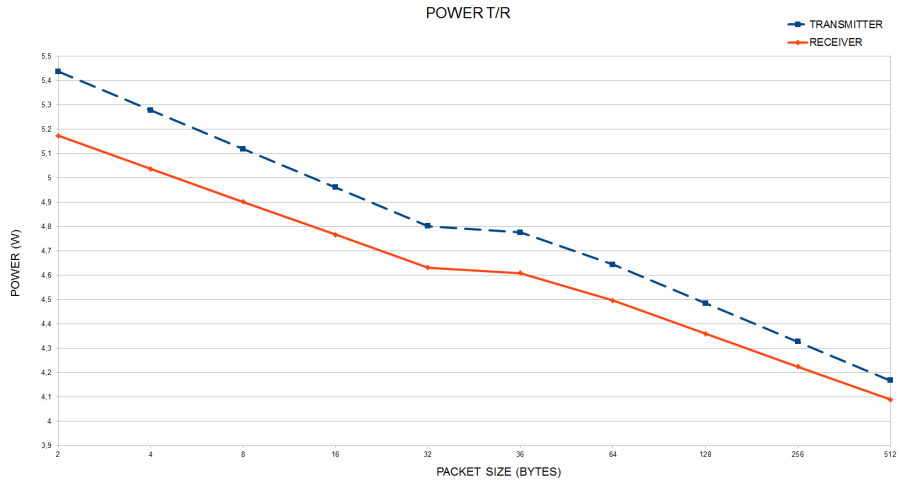


Figure 9: Average power consumption at the wireless sensor node as a function of the datagram size for a constant traffic generation rate of 10Mb/s.

verifiable multilateration technique was used in this scenario.

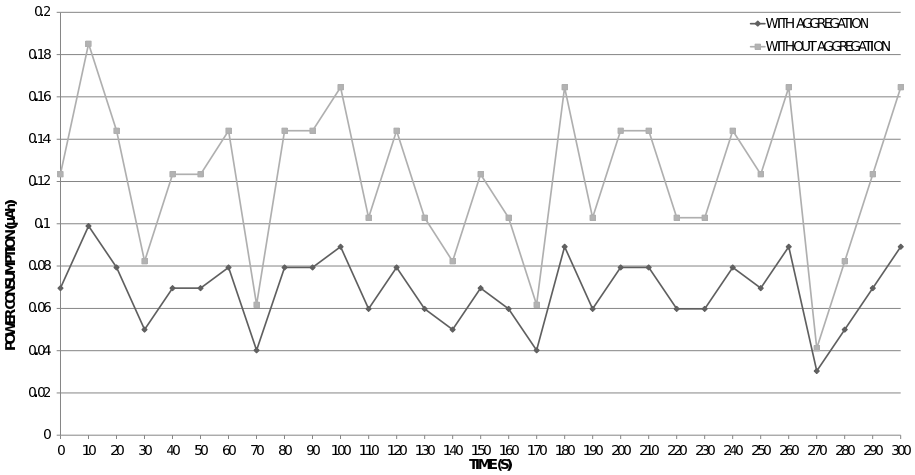
In the envisioned application, the WSN is required to monitor the temperature of a certain area, and as a result, each sensor periodically generates a random temperature sample uniformly distributed in the range $[28, 32]$. Period is set to 5 seconds.

Table 3 and 4 respectively report the number of packets and bytes sent at each hop of the network. As in [4], we consider three scenarios: (i) all sensor nodes reply; (ii) 90% of the nodes replies; and (iii) only 70% of the sensor nodes replies. *Cluster heads* (i.e. mesh routers) do not generates any sample, moreover, we assume that the distribution of non–responding nodes is uniform across all the clusters.

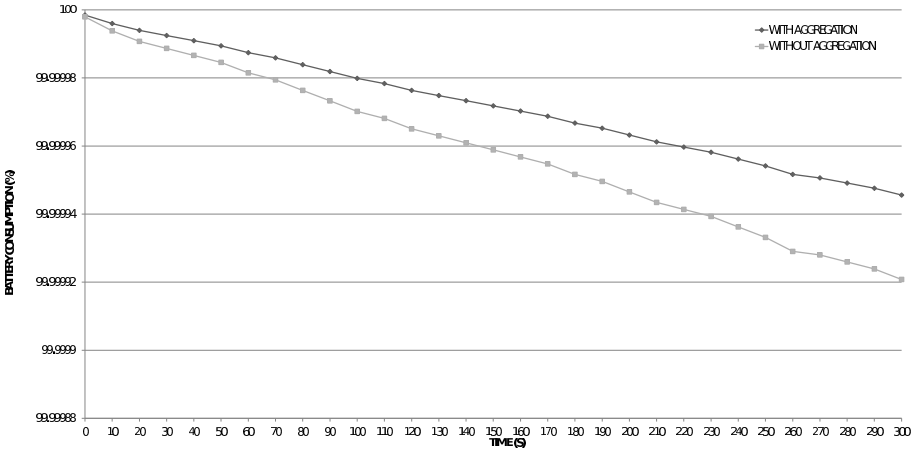
As it can be seen, in the *No-Agg* scenario, nodes that are closer to the *Sink* transmit an amount of data that is significantly higher (see *Hop 4* in the tables) than the data transmitted by the previous *Cluster Heads*. On the other hand, in the *Agg* scheme, the number of transmissions is constant while the amount of bytes exchanged at each hops increases. Such a behavior is due to the *ids* of the non–responding nodes that need to be appended to the aggregated samples being transmitted. Such a list becomes larger and larger as the sample get closer to the *Sink*.

6. Related work

State-of-the-art solutions for secure data aggregation can be classified as hop–by–hop data aggregation and end–to–end data aggregation. In the former ap-



(a) Power consumption.



(b) Battery lifetime.

Figure 10: Power consumption and battery lifetime for a sensor node in a cluster.

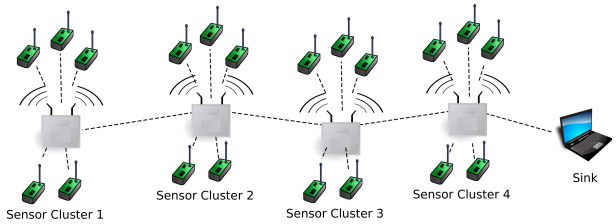


Figure 11: The linear network topology exploited during our study.

proach, data is encrypted by the sensing nodes and decrypted at each hop before being delivered to the *Sink*. In the latter approach, data is decrypted only by the

Table 3: Number of packets relayed at each hop.

Hops	**No–Agg**	**Agg**	**Agg (90%)**	**Agg (70%)**
1	10860	180	180	180
2	21660	180	187	193
3	32520	180	188	193
4	43380	181	188	193

Table 4: Number of bytes relayed at each hop.

Hops	**No–Agg**	**Agg**	**Agg (90%)**	**Agg (70%)**
1	434400	7200	8552	10628
2	866400	7200	10784	16036
3	1300800	7200	12532	20096
4	1735200	7240	14086	24084

Sink.

Different hop–by–hop solutions [2, 15, 16, 3] assumes that data security is guaranteed by means of some key distribution schema. For example SEDAN [3] proposes a secure hop–by–hop data aggregation protocol, in which each node can verify the integrity of its two hops neighbors' data. SEDAN [3] provides a totally distributed scheme to guarantee data integrity. The SEDAN performance, evaluated by means of ad-hoc simulation, shows a better behavior than other solutions, i.e., SAWAN [2], in terms of overhead and mean time to detection. Nevertheless, all hop–by–hop secure data aggregation solutions are vulnerable to attacks at the intermediate nodes, that can be tampered, leaving the attackers with complete access to the sensor readings.

In [15] the authors tackle the problem of enabling secure data aggregation and verification in sensor networks. The authors divide the network in clusters and assume that up to a certain number of nodes in a cluster can be compromised. The proposed solution consists of two parts: a key establishment protocol that generates a different key for each cluster where each node knows only a share of the cluster key; and a secure data aggregation protocol that ensures that the *Sink* does not accept faulty readings. A protocol for the efficient computation of statistical values such as the median and the average of the measurements performed by a WSN is presented in [16]. The protocol can also estimate the network size and find the minimum and maximum sensor reading. In their work the authors propose the *aggregate–commit–prove* framework where aggregators not only perform the aggregation tasks, but must also prove that they perform these tasks correctly enabling the user to verify that answer given by the aggregators is a good approximation of the true value even when the aggregators and/or

a fraction of the sensor nodes may be corrupted.

End–to–end techniques, such as [4, 17, 18, 19, 20, 6], overcome this limitation by requiring all the nodes to share an encryption key only with the *Sink* possibly using novel distribution schemes [21, 22, 23].

Particularly, SeDap [20], one of our previous works, addresses the privacy as well as security aggregation issues adopting an end-to-end additively homomorphic encryption. But this work does not adopt a hybrid architecture and the evaluation of node reputation is not performed. An alternative approach is represented by the use of public–key encryption scheme, such as the one presented in [18]. The drawback of this solution is represented by the high computational requirements imposed by public–key schemes. DyDAP [6] is another our previous work that combines privacy management with an original end–to–end aggregation algorithm. By using simulation tools, we demonstrate that DyDAP can effectively reduce the load in congested WSN while at the same time guaranteeing anonymity and data integrity. In [17, 19] the authors present an end–to–end encryption scheme for reverse multicast traffic, i.e. traffic between sensor nodes and the *Sink*. The proposed approach allows nodes to perform data aggregation operations by operating only on the ciphertext, which in time provides the advantage that intermediate aggregators do not have to carry out costly decryption and encryption operations, and thus, do not require to store sensitive cryptographic keys.

In [21] the authors tackle the problem of designing a clustered sensor networks able to isolate the effects of malicious nodes, i.e. captured nodes, to specific clusters or subgroups. The proposed scheme can maintain flexibility in providing different security concerns for different sensor subgroups. Similar objectives are pursued by the authors of [22] that aim at providing any pair of nodes in a sensor network with the possibility to establish a confidential and secure communication channel while loading each sensor with a small set of keys. The solution is based on two protocols. The former is secure with a fixed probability and that is used for the initial handshaking procedures. The latter protocol instead has a level of security that can be traded off with the overall communication overhead.

Two cooperative protocols (CoMAC and ExCo) are presented in [23]. The two schemes use standard (and inexpensive) symmetric cryptographic primitives coupled with key evolution and few messages exchange. The authors evaluate the two solutions using simulations and show that network designers can carefully select the right scheme and tune appropriate system parameters in order to achieve the desired level of robustness and overhead.

As opposed to the aforementioned solutions, our work exploits an hybrid sensor/mesh network architecture where an homomorphic encryption scheme is implemented by the sensor nodes, while data aggregation operations are performed by mesh routers that are not required to know the actual content of the message

being processed. Our architecture is capable of providing data confidentiality and integrity while, at the same time, reducing the amount of traffic exchanged over the network and thus the overall power consumption.

In [24] a trade–off between data protection and feasibility/complexity is presented. In particular, the authors exploit data aggregation and secure localization in order to centrally (i.e., at the *Sink*) compute an overall assessment of the data quality. Similarly, in [10, 9] the authors exploit a technique known as *Verifiable Multilateration* in order to improved the trustworthiness of sensor nodes' localization information. The authors show that when the verifiers play a mixed strategy, the malicious node can masquerade as non–malicious with very low probability. In this work we combined the aforementioned secure localization mechanism with an hybrid data aggregation scheme for wireless sensor/mesh networks. The hybrid nature of our architecture allows sensor nodes to use their resources, i.e. their battery, only to implement sensing and data forwarding functionalities, leaving mesh routers in charge for secure data aggregation and secure node localization. Localization information are exploited for evaluating node reputation.

The problem of collaborative data aggregation and data privacy is jointly addressed by the authors of [25]. Two algorithms are proposed: CPDA and SMART. The former exploits a clustering protocol in order to reduce the communication overhead. The latter employs data slicing techniques in order to securely distribute the data to be aggregated to the nearest sensor nodes for aggregation. Our architecture introduces on the one hand a sharing of tasks between sensor nodes and mesh routers for what concern data aggregation and security, and on the other hand exploits the concept of reputation for improving data quality.

In [26] the authors propose a set of secret perturbation schemes that can effectively address sensor data confidentiality issues without losing the bandwidth efficiency gains delivered by the concept of additive data aggregation. The proposed approach requires the *Sink* to share a secret with each sensor node. A sensor node that wants to report some sensory data to its *Sink*, first sums the original data with the secret and then transmit the result of the operation to the *Sink*. As opposed to our solution, this work addresses only data confidentiality without tacking the challenge of node reputation that improves the security level of the whole network.

7. Conclusion

In this paper we presented DARE an architecture addressing both node reputation and *run-time* data trustworthiness. DARE is based on an hybrid wireless sensor/mesh networks architecture which allows to allocate computationally intensive tasks such as the secure localization technique based on verifiable multilateration and data aggregation to the mesh routers, leaving sensor nodes in

charge of the mere data gathering functionalities.

Simulations results show a significant improvement in terms of both amount of data exchanged over the wireless medium and power consumption at the wireless sensor nodes. Moreover, we found a clear correlation between the number of verifiers deployed in the network and the number of messages that can be aggregated. Finally, a real–world prototype has been implemented and tested in order to verify the suitability of our architecture to a real use case, collecting and processing the mean and the average value of the temperature measured by a WSN.

As a future work we plan to further investigate the trade-offs between number of verifiers and the performance of the network in terms of aggregated message and thus energy consumption. The investigation of smart techniques for deploying verifiers across the network is also considered and research direction worth pursuing. Furthemore, the application of multimedia data secure aggregation techniques to our architecture is under study. Finally, in order to better characterize the security profile of the sensor nodes and the related sensing data trustworthiness additional metrics are under investigation.

References

[1] S. Capkun, J.-P. Hubaux, Secure positioning of wireless devices with application to sensor networks, in: Proc. of IEEE INFOCOM, Miami, Florida, USA, 2005.

[2] L. Hu, D. Evans, Secure data aggregation in wireless sensor networks, in: Proc. of IEEE WSAAN, Orlando, FL, USA, 2003.

[3] M. Bagaa, N. Lasla, A. Ouadjaout, Y. Challal, Sedan: Secure and efficient protocol for data aggregation in wireless sensor networks, in: Proc. of IEEE LCN, Dublin, Ireland, 2007.

[4] C. Castelluccia, E. Mykletun, G. Tsudik, Efficient aggregation of encrypted data in wireless sensor networks, in: Proc. of MobiQuitous, San Diego, CA, USA, 2005.

[5] I. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Elsevier Computer Networks 47 (4) (2005) 445 – 487.

[6] S. Sicari, A. Grieco, G. Boggia, A. C. Porisini, Dydap: A dynamic data aggregation scheme for privacy aware wireless sensor networks, Elsevier Journal of Systems & Software 85 (1) (2012) 152 – 166.

[7] A. Coen-Porisini, S. Sicari, Improving data quality using a cross layer protocol in wireless sensor networks, Computer Networks 56 (17) (2012) 3655 – 3665. doi:10.1016/j.comnet.2012.08.001.

[8] A. Coen-Porisini, S. Sicari, Cross layer data assessment in wireless sensor networks, in: Proc. of SENSORNETS 2012, Rome, Italy, 2012.

[9] M. Monga, S. Sicari, On the impact of localization data in wireless sensor networks with malicious nodes, in: Proc. of ACM SIGSPATIAL, New York, NY, USA, 2009.

[10] N. Gatti, M. Monga, S. Sicari, Localization security in wireless sensor networks as a non-cooperative game, in: Proc. of IEEE ICUMT, Moscow, Russia, 2010.

[11] R. Riggio, T. Rasheed, S. Sicari, Performance evaluation of an hybrid mesh and sensor network, in: Proc. of IEEE Globecom, 2011.

[12] K. Gomez, R. Riggio, T. Rasheed, D. Miorandi, F. Granelli, Energino: a Hardware and Software Solution for Energy Consumption Monitoring , in: Proc. of IEEE WinMee, Paderborn, Germany, 2012.

[13] I. Akyildiz, I. Kasimoglu, Wireless sensor and actor networks: research challenges, Elsevier Ad Hoc Networks 2 (4) (2004) 351 – 367.
[14] R. Riggio, N. Scalabrino, D. Miorandi, F. Granelli, Y. Fang, E. Gregori, I. Chlamtac, Hardware and software solutions for wireless mesh network testbeds, IEEE Communication Magazine 46 (6) (2008) 156 – 162.
[15] A. Mahimkar, T. Rappaport, Securedav: A secure data aggregation and verification protocol for sensor networks, in: Proc. of IEEE Globecom, Dallas, Texas, USA, 2004.
[16] B. Przydatek, D. Song, A. Perrig, Sia: Secure information aggregation in sensor networks, in: Proc. of ACM SenSys, Los Angeles, California, USA, 2003.
[17] J.Girao, D. Westhoff, M. Schneider, Cda: concealed data aggregation for reverse multicast traffic in wireless sensor networks, in: Proc. of IEEE ICC, Seoul, Korea, 2005.
[18] E. Mykletun, J. Girao, D. Westhoff, Public key based cryptoschemes for data concealment in wireless sensor networks, in: Proc. of IEEE ICC, Istanbul, Turkey, 2006.
[19] D. Westhoff, J. Girao, M. Acharya, Concealed data aggregation for reverse multicast traffic in sensor networks: Encryption, key distribution, and routing adaptation, Mobile Computing, IEEE Transactions on 5 (10) (2006) 1417 –1431. doi:10.1109/TMC.2006.144.
[20] A. Coen-Porisini, S. Sicari, Sedap: Secure data aggregation protocol in privacy aware wireless sensor networks, in: Proc. of ICST S–Cube, Miami, FL, USA, 2010.
[21] L. Eschenauer, V. D. Gligor, A key-management scheme for distributed sensor networks, in: Proc. of ACM CCS, Washington, DC, USA, 2002.
[22] R. D. Pietro, A. Mei, L. V. Mancini, Random key assignment for secure wireless sensor networks, in: Proc. of ACM SASN, Fairfax, VA, USA, 2003.
[23] R. D. Pietro, C. Soriente, A. Spognardi, G. Tsudik, Collaborative authentication in unattended wsns, in: Proc. of ACN WiSec, Zurich, Switzerland, 2009.
[24] M.Monga, S. Sicari, Assessing data quality by a cross-layer approach, in: Proc. of IEEE ICUMT, St. Petersburg, Russia, 2009.
[25] W. He, X. Liu, H. Nguyen, K. Nahrstedt, T. Abdelzaher, Pda: Privacy-preserving data aggregation in wireless sensor networks, in: Proc. of IEEE INFOCOM, Anchorage, AL, USA, 2007.
[26] T. Feng, C. Wang, W. Zhang, L. Ruan, Confidentiality protection for distributed sensor data aggregation, in: Proc. of IEEE INFOCOM, Phoenix, AZ, USA, 2008.