SeDAP: Secure Data Aggregation Protocol in Privacy Aware Wireless Sensor Networks

Alberto Coen Porisini and Sabrina Sicari

Dipartimento di Informatica e Comunicazione
Università degli studi dell'Insubria
Via Mazzini, 5-21100 Varese-Italy
{alberto.coenporisini,
sabrina.sicari}@ uninsubria.it

Abstract. Wireless Sensor Networks are characterized by very tight energy requirements and therefore they impose strict constraints on the amount of data that can be transmitted by sensors nodes. In order to achieve such a goal many data aggregation algorithm have been defined. However, in order to ensure a broad deployment of the innovative services delivered by WSNs, strict requirements on security and privacy must be also satisfied. The aim of this paper is to present an integrated framework, named SeDAP, that deals with end-to-end data aggregation and privacy issues.

Keywords: WSN, privacy, end-to-end secure data aggregation.

1 Introduction

Wireless Sensor Networks (WSN) technologies support data collection and distributed data processing by means of very small sensing devices [1], with limited computation and energy capabilities. WSN are used in many contexts, such as telemedicine, surveillance systems, assistance to disabled and elderly people, environmental monitoring, localization of services and users, industrial process control, and systems supporting traffic monitoring/control in urban/suburban areas, military and/or anti-terrorism operations.

An important goal when designing WSN systems is the reduction of the need for transmission since from a power consumption perspective, data transmission is a very expensive operation. One solution consists in using proper aggregation algorithms (e.g., see [2], [3], [4], [5]) that can reduce significantly the number of bytes exchanged across the WSN. In fact, in many situations, what is needed are aggregated measures, such as the average temperature of a region, the average humidity, and so on. Thus, network processing capabilities and data-aggregation are key features of a WSN, which can greatly improve energy efficiency by reducing data going through the wireless channel [3], [4].

Another important issue in WSN is represented by privacy that may be violated by tampering of sensors and/or traffic due to the nature of the wireless channel and its

G. Parr and P. Morrow (Eds.): S-Cube 2010, LNICST 57, pp. 135–150, 2011.

deployment in uncontrolled environments. Thus, privacy aware mechanisms are crucial for several WSN applications such as localization and telemedicine. Moreover, it is necessary to take into account privacy also in those application contexts in which data referring to individuals are not directly handled by the WSN. For example, in home networks, sensor nodes may collect a large amount of data that may reveal habits of individuals, violating in this way their privacy. Among the different aspects characterizing privacy, anonymity is an important requirement for a privacy aware system that aims at protecting the identity of the individuals whose data are handled by the system.

In this paper we take into account both data aggregation and privacy issues following the modeling approach proposed in [7], [8] and [9] in order to define an integrated solution that considers a solid privacy management policy coupled with an aggregation algorithm [2]. The aggregation process, which merges spatial correlated data and works on encrypted information, involves only linear operations and allows the sink node to estimate the confidence level of aggregated data.

The model is defined in UML [10], [11] and represents a general schema that can be easily adopted in different contexts. It introduces concepts, such as nodes, data, actions, that are needed to define a privacy policy along with the existing relationships among them.

The main objectives fulfilled by our approach are: (i) anonymity management; (ii) data integrity check; (iii) data aggregation to reduce the network load; (iv) end-to-end secure data aggregation.

The rest of the paper is organized as follows. Section 2 introduces the foundations for modeling privacy in the context of WSN and presents a short overview of the conceptual model. Section 3 describes the reference scenario and the adopted end-to-end secure data aggregation algorithm. Section 4 introduces SeDAP the proposed framework, integrating privacy management policies and data aggregation. Section 5 shows some performance evaluation of the proposed algorithm. Section 6 presents some related works. Finally, Section 7 draws some conclusions and provides hints for future works.

2 Privacy Model

A privacy policy defines the way in which data referring to individuals can be collected, processed, and diffused according to the rights that individuals are entitled to. The rest of the paper adopts the terminology introduced by the EU directive [12]. Notice that, since the proposed terms are general, i.e., they are not dedicated to a specific type of network, it is necessary to refine them in order to provide the concepts needed for supporting the definition of privacy mechanisms in WSN communications. In the following, a short overview of the conceptual model for privacy policies is illustrated. The structural aspects are defined using UML classes and their relationships. Figure 1 depicts a class diagram that provides a high level view of the basic structural elements of the model.

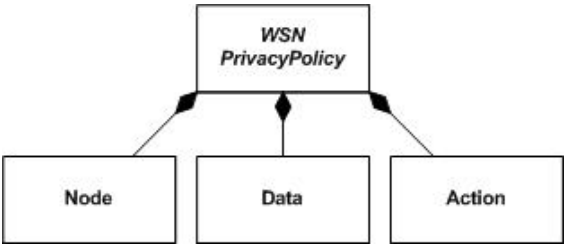

Fig. 1. A WSN Privacy Policy

A *WSN Privacy Policy* is characterized by three types of classes: *Node*, *Data, and Action*. Nodes interact among them inside the network in order to perform some kind of actions on data. Thus, an instance of *WSN PrivacyPolicy* is characterized by specific instances of *Node*, *Data*, and *Action*, and by the relationships among such entities. Now, let us focus on the classes introduced by the diagram.

1) Node. It represents a member of the network either interested in processing data or involved by such a processing. Nodes are characterized by functions and roles (see Figure 2).

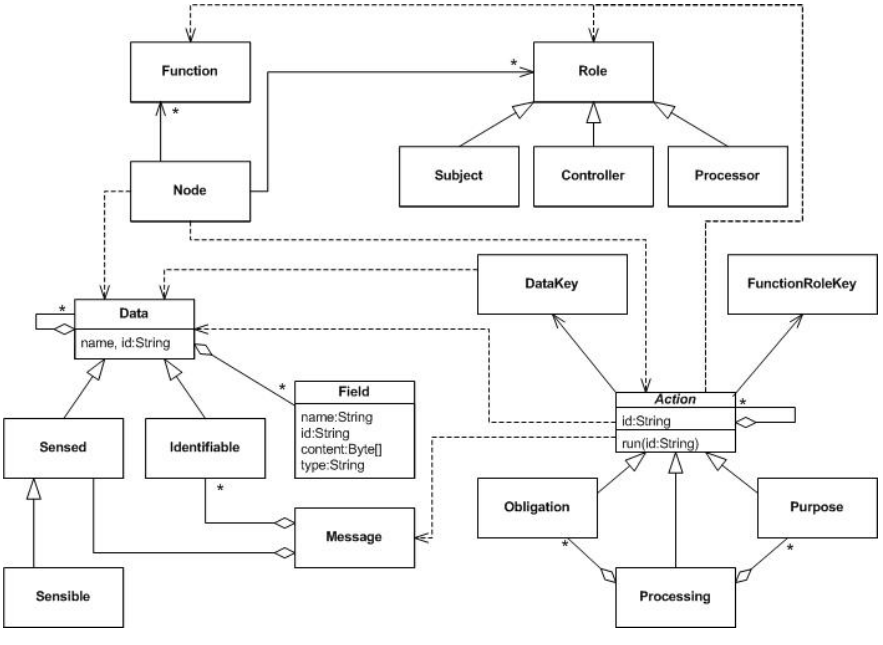

Fig. 2. UML Model

More specifically:
- *Role* [13] is a key concept of this approach; in fact, nodes are characterized depending on the role they play with respect to privacy. Three distinct classes

represent the different roles: Subject, which is a node that senses the data; Processor, which is a node that processes data by performing some kind of action on them (e.g., transmission, forwarding, aggregation, etc.); Controller, which is a node that verifies the actions executed by processor nodes.

- *Function* represents the task performed by a Node within the network in which it operates (e.g., data sensing, message transmission, message forwarding, data aggregation, etc.).

- *Data*. It represents the information referring to subjects that can be handled by processors. Data is extended by means of: *Identifiable* data (e.g., node identifier) which represent the information that can be used to uniquely identify nodes; *Sensitive* data (e.g., health related data) which represent information that deserve particular care and that should not be freely accessible; *Sensed* data (e.g., temperature, pressure) which contain information that are sensed by the nodes of the network.

In the WSN context, sensitive data may be considered an extension of sensed data, i.e., they are sensed data related to individuals which require a particular care. For instance, in telemedicine applications a sensitive datum is the temperature which is sensed by nodes positioned on the body of patients. Notice that in the context of WSN also common sensed data deserve particular care. For example, consider a wireless meter reading system used to monitor the temperature and pressure of different rooms of the building where it is installed; such a system comprises several sensor units which communicate information on the current temperature, barometric pressure, and humidity of the rooms where they are positioned. Although the data sensed by the nodes of the system cannot be classified as sensitive, they can be used to reveal information on personal habits of the people who live in the interested building. As an example, slight increments of temperature or of humidity may reveal the presence of one or more person in a room. By analyzing such data, it is possible to infer periods of the day or of the week during which the building is empty.

Data is a complex structure composed of basic information units, named *Fields*, each of which represents a partial information related to the whole data structure.

- *Message.* It represents the basic communication unit exchanged by the nodes of the network. It contains *Identifiable* information concerning the nodes involved in the communication and *Sensed* data.

- *Action.* It represents any operation performed by *Node*. It is extended by *Obligation*, *Processing*, and *Purpose*. Moreover, each action can be recursively composed of other actions. Since in a privacy aware scenario a processing is executed under a purpose and an obligation, *Processing* specifies an aggregation relationship with *Purpose* and *Obligation*.

Notice that while in general for each function there may be defined several actions that can be performed, in the context of WSN usually each function corresponds to one action. To guarantee the confidentiality and integrity of data as well as to assure that only authorized nodes are allowed to access such data and execute actions, our model introduces encryption mechanisms. More specifically, two classes representing encryption keys, named *DataKey* and *FunctionRoleKey*, are introduced. The former class is used for the definition of encryption mechanisms to protect the data content of messages; whereas the latter is used for defining mechanisms to assure that message communication and data handling are executed only by authorized nodes. Each node

of the network owns a different *DataKey* used to encrypt the data content of the messages. Each node also owns multiple *FunctionRoleKey* that are used to constrain which nodes can execute specific actions on data. *Actions* are expressly built to be executed by nodes that belong to a given function-role combination.

Since a node may play different functions and roles, it may own multiple function-role keys, one for each pair of function-role. The system modeler is allowed to use the key generation algorithm and the encryption algorithm that he/she considers the most suitable for the application domain.

3 The Scenario

We consider a dense network composed of N nodes; each node senses a given type of data (e.g., temperature, pressure, brightness, position, and so on). Nodes can exchange messages and all sensed data (possibly aggregated) are directed to the sink. Each node directly communicates with its closer neighbors (at one hop distance). The broadcast nature of wireless channels enables a node to determine, by overhearing the channel, whether its messages are received and forwarded by its neighbors [14]. Each node of the network is characterized by a label n in order to unambiguously identify the node in the network.

Each node owns different types of keys each of which corresponds to a given Function-Role pair. We identify the following Function-Role pairs: Sensing-Subject (SS); Authenticator-Processor (AP);Transmitter-Processor (TP); Notifier-Controller (NC). Thus, each node owns four keys, one for each pair Function-Role. Keys are denoted by $k(n, fr)$, where n is the node label and $fr \in$ {SS,AP, TP,NC} is the Function- Role played by node n.

Notice that, we assume that keys are pre-shared in the nodes and that each node contains a table in which it stores the last sent messages. The usefulness of this table will be clarified later on.

Each node, in order to achieve end-to-end security to data that are aggregated inside a WSN, adopts the algorithm of Castelluccia et al. [2] which is based on a simple and secure additively homomorphic stream cipher that allows efficient aggregation of encrypted data. The new cipher only uses modular additions and is therefore very well suited for CPU-constrained devices like sensors. 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. Homomorphic encryption schemes are especially useful in scenarios where someone who does not have decryption keys needs to perform arithmetic operations on a set of ciphertexts.

For reader convenience, we will briefly sketch the additively homomorphic encryption scheme proposed in [2] to show how it works during aggregation.

Each node n_i represents its data as an integer number d_i and let $k(n_i,SS)$ be a randomly generated keystream, *Enc* and *Dec* be the encryption and decryption function respectively, and M a large enough integer number. Then, the encrypted ciphertext c_i is given by:

$$c_i = Enc(d_i; k(n_i, SS);M) = (d_i + k(n_i, SS)) \bmod M \quad (1)$$

The sensor then forwards the ciphertext to its parent, who aggregates all the c_i received from its children:

$$c = \sum_{i=1}^{z} c_i \,(\mathrm{mod}\, M) \tag{2}$$

The cleartext aggregated data d can then be obtained by:

$$d = Dec(c;k;M) = (c-k) \bmod M \tag{3}$$

$$where\ k = \sum_{i=1}^{z} k(n_i, SS)$$

Due to the commutative property of addition, the above encryption scheme is additively homomorphic. In fact, if $c_1 = Enc(d_1; k(n_1, SS);M)$ and $c_2 = Enc(d_2; k(n_2, SS);M)$ then $c_1 + c_2 = Enc(d_1 + d_2; k(n_1, SS) + k(n_2, SS);M)$.

Note that if α different ciphers c_i are added, then M should be larger than $\sum_{i=1}^{\alpha} d_i$, otherwise correctness is not provided. In fact, if $\sum_{i=1}^{\alpha} d_i$ is larger than M, decryption will results in a value d^* that is smaller than M. In practice, if $p = \max\{d_i\}$, then M should be selected as ($M = 2^{\log_2(p \cdot \alpha)}$). 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.

4 SeDAP: The Proposed Solution

In this section, the novel integrated framework SeDAP to realize end-to end data aggregation scheme with privacy capabilities in WSN is described. In particular, we address the following issues: (i) data integrity; (ii) anonymity; (iii) energy efficient WSN usage; and (iv) end-to-end secure data aggregation.

4.1 Message Structure

To exploit the benefits derived by the adoption of SeDAP, which satisfies both end-to-end data aggregation and anonymity requirements, network messages have to be suitably structured. More specifically, a message contains data that, according to the conceptual model, may be classified as *identifiable* and *sensed*. *Identifiable* data includes the information that can be used to identify a node. *Sensed* data includes all information sensed by the nodes, such as the environmental temperature, pressure, and so on. A message refers to a single transmission hop between adjacent nodes. It is identified by the notation $msg_{n,q}$ where n identify the node that generated and transmitted the message, while q identifies the message among those generated by

node *n*. The pair *(n,q)* unambiguously identifies the message among those transmitted in the network.

A sensed data before reaching the sink passes through different nodes of the network (multi-hop communication) by means of different messages. To guarantee the integrity and confidentiality of the end-to-end communication, we propose a message structure that keeps track of the last two hops of the transmission. In this way it is possible to implement a simple enforcement schema that checks the integrity of the data content of the message.

Combining the requirements of both anonymity and data aggregation, a generic SeDAP message $msg_{n,q}$, is a tuple

$msg_{n,q}$=<*current, previous, subject, sensing−identifier, mistaking−identifier, error−flag, data, id-list*>

where:

- *current*: is a couple <nc, qc>, which unambiguously identifies the current message among the ones transmitted within the network. This field is ciphered.
- *previous*: is a couple <np, qp>, which includes np, the identifier of the node that operated the second last forwarding of the sensed data contained in the current message, and qp, the identifier used by np to identify such a message. This field is ciphered.
- *subject*: is a couple <ns, qs> where ns is the identifier of the node subject which originally sensed the data or the aggregator node which aggregated the data. Whereas, qs is the message identifier used by such a node(aggregator or subject) for the message that started the communication of the sensed data towards the *sink*. Notice that in case of error notification (see Reception and Integrity Verification Protocol) this field identifies the node that reveals the error. This field is ciphered.
- *sensing-identifier*: is a tuple <nsi, qsi> that in case of error notification contains the identifier of the node that sensed or aggregated the correct data and the identifier of the message transmitted by such a node. This field is ciphered.
- *mistaking identifier*: is a tuple <nmi, qmi>, which contains the identifier of the node that generated the error and the identifier of the message containing the error transmitted by such a node. This field is ciphered.
- *error-flag*: represents an error code, which indicates if an anomaly was identified in the message content. This field is in clear.
- *data*: includes the ciphered data c either sensed or aggregated by the subject node or the aggregator node, respectively.
- *identifier list*: is the list of the encrypted node identifiers used by the sink in the decryption process for identifying the nodes and then the keys that handle the data.

Notice that *sensing identifier*, *mistaking identifier* are used only in case of error notification, i.e., when *error flag* is set to 1, as described below.

4.2 System dynamics

System dynamics are described by means of the following protocols:

- *Sensing,* which defines the actions that a node of the network executes to sense data and to communicate such data to the other nodes of the network.

- *Message Reception and Integrity Verification,* which defines the actions that a node should perform to both forward data received from other nodes and verify the integrity of the messages transmitted across the network.
- *Data Aggregation,* which defines the action that a node of the network executes to aggregate sample encrypted sensed data.

4.2.1 The Sensing Protocol

The operations are described below step by step.

1. *Data sensing*. The node n_c senses a data *d* from the environment where it is located. The node plays the role of *subject* and the function of *sensing*.
2. *Data encryption*. The node encrypts the sensed data *d* by using its sensing-subject key $k(n_c, SS)$. [1] The resulting output is c that is equal to $Enc(d; k(n_c, SS)$.
3. *Message identifier generation*. The node generates an identifier q_c for the message that has to transmit to the sink $<n_c, q_c>$
4. *Identifiable data encryption*. The node encrypts the generated identifier by using its personal *transmitter-processor* key, $k(n_c, TP)$. As a result, we have the content $Enc(<n_c, q_c>; k(n_c, TP))$.
5. *Message structuring*. A new message $msg_{nc,qc}$ is generated starting from the resulting outputs of the previous steps 2, 3 and 4, with the following structure:

- *current* is set to $Enc(<n_c, q_c>; k(n_c, TP))$;
- *previous* is initialized to an empty string, because this is the first transmission and no forwarding has been executed yet;
- *subject* is set to $Enc(<n_c, q_c>; k(n_c, TP))$ since the current transmitter is the subject itself;
- *sensing-identifier* is an empty string because the message is not an error notification message;
- *mistaking-identifier* is an empty string because the message is not an error notification message;
- *error -flag* is set to 0 because there is no error;
- *data* is set to $c = Enc(d; k(n_c, SS))$;
- *identifier list* is updated with the encrypted identifier of the node n_c. Notice that this field is composed by only one identifier, equal to the field current, since this is the first transmission and no forwarding has been executed yet;

6. *Message storing*. The node stores the content of the fields *data* and *subject* in its local table. It uses the content of the field current of the message $msg_{nc,qc}$ as the hash key for the sensed data that needs to be stored.
7. *Message queuing*. The message is put in the transmission queue.

4.2.2 Message Reception and Integrity Verification Protocol

The operations executed by each node when a packet is received and then forwarded are the following:

1. *Role check*. The node n_c analyses the received message $msg_{np,qp}$ to figure out what type of action it has to execute on the contained data. In particular, it looks for the message among those stored in the local table by using the content of *previous* field of the received message as hash key.

[1] The Sensing-Subject key is equivalent to the DataKey defined in the conceptual model.

If the message is not found, this means that it was not previously transmitted by the node. In this case, the node changes its current function and role, i.e., it has to play the role of *processor* and the function of *transmitter.* Therefore the node executes the following steps:

2a. *Message identifier generation*. The node generates an identifier q_c for the message that has to put in the transmission queue $<n_c, q_c>$.
3a. *Identifiable data encryption*. The node encrypts the generated identifier by using its personal *transmitter-processor* key, $k(n_c, TP)$.
4a. *Message structure*. The new message $msg_{nc,qc}$ has the field *current* equal to $Enc(n_c, q_c ; k(n_c, TP))$.Obviously, the *previous* field is equal to the *current* field of the received $msg_{np,qp}$. In the identifier list is added the identifier of the current node. The other fields remain unchanged.
5a. *Message storing*. The node stores the content of the fields *data* and *subject* in its local table. It uses the content of the field current of the message $msg_{nc,qc}$ as the hash key for the sensed data that have to be stored.
6a. *Message queuing*. The message is put in the transmission queue.

Otherwise, that is if the message is found, it means that it was originally transmitted by node itself. In this case, the node changes its current function and role, i.e., it has to play the role of *controller* and the function of *notifier* to verify the integrity of the previously transmitted message. Hence, the node compares the content of field *data*, of the received message with the information retrieved from its table. If the information match, this means that the *controller* is sure that the node from which it received the message preserved the integrity of the data, position and weight content. In this case, no additional action is performed by the node.

If the content of field *data* is different from the ones extracted from the local table or no data entry corresponds to the search key. This means that something wrong happened. In this case, the node generates a new message as described in what follows, in order to notify the sink that a corrupted message is spreading through the network:

2b. *Message identifier generation*. The node generates an identifier q_c for the message that has to put in the transmission queue $<n_c, q_c>$
3b. *Identifiable data encryption*. The node encrypts the generated identifier by using its personal *Transmitter-Processor* key, $k(n_c, TP)$.
4.b *Message structure*. The new message $msg_{nc,qc}$ has the field *current* equal to $Enc(n_c, q_c ; k(n_c, TP))$. The *previous* field is empty to avoid the creation of some loops with the malicious node and the spreading of different and opposite error messages; *subject*, which is equal to the field *current* to specify identifiable information of the node that retrieved the error (since such a node is the current one, the content of *subject* is equal to *current*); *sensing identifier* is set to the value of field *subject* of the node that senses/ aggregates the correct data that is stored in the node local table; *mistaking identifier* is equal to the field *current* of the received message in order to provide information about the node that makes the mistake, *error flag* is set to 1 to indicate that the current message is an error message. The fields *data* is equal to the content stored in the local table of the node and are encrypted with the *Notifier-Controller* key of the current node $k(n_c,NC)$. The *identifier list* is updated with the identifier of the current node.

5b *Message storing*. The node stores the content of the fields *data* and *subject* in its local table. It uses the content of the field current of the message $msg_{nc,qc}$ as the hash key for the sensed data that have to be stored.
6b. *Message queuing*. The message is put in the transmission queue.

4.2.3 Data Aggregation Protocol

The data aggregation is periodically triggered by each node. It involves the following steps.

1. *Error Check*: the node checks the field *error flag* of the message in the transmission queue. If the field is set to 1, this means that the message is an error notification and no aggregation operations is possible to perform on the contained data. The message is transmitted as it is. Otherwise if the error flag is set to 0 then the aggregation procedure starts.
2. *Ciphered data aggregation*. The data selection procedure iteratively operates to arrange enqueued message. The data contained in all messages in the transmission queue are aggregated in a single message. More specifically, the ciphered data c_i received from the children nodes, that are respectively encrypted by the Sensing-Subject key of each child, are aggregated following the equation (2) (see Section 4). Notice that the aggregation process is performed without any knowledge of the keys from the aggregator node.
3. *Message identifier generation*. The node generates an identifier q_c for the message that has to put in the transmission queue $<n_c, q_c>$.
4. *Identifiable data encryption*. The node encrypts the generated identifier by using its personal transmitter-processor key, $k(n_c, TP)$.
5. *Message structure*. The new message $msg_{nc,qc}$ has the field *current* equal to $Enc(n_c, q_c ; k(n_c, TP))$. The field *previous* is initialized to an empty string because this is the first transmission of the aggregated data and no forwarding has been executed yet. The field *subject* is set to $Enc(n_c, q_c ; k(n_c, TP))$; notice that the field *subject* is equal to the field *current* because the aggregator is the generator of the aggregated data. *data* is set to $c = \sum_{i=1}^{z} c_i \pmod{M}$ according to equation (2) in Section 3. Finally, the field *identifier list* is updated with the identifier of the current node. Then, the node performs the message storing and message transmission procedures described as the other protocols .

5 Performance Evaluation

In order to evaluate the efficiency of the proposed solutions some simulations have been conducted, by means of Omnet ++ [16].We consider a wireless sensor network, which measures the temperature of a given environment. The tests compare the behaviour of a network that uses SeDAP with the behaviour of a network that adopts only a secure end-to-end data aggregation, such as Castelluccia et al.[2]. The simulations confirm the expected results in terms of number of transmitted messages and transmitted bytes; more specifically in order to guarantee anonymity and integrity, as shown in Figures.3 and 4, where Gen[] and Aggregator[] represent the nodes that sense single data, and the nodes performing aggregation, respectively.

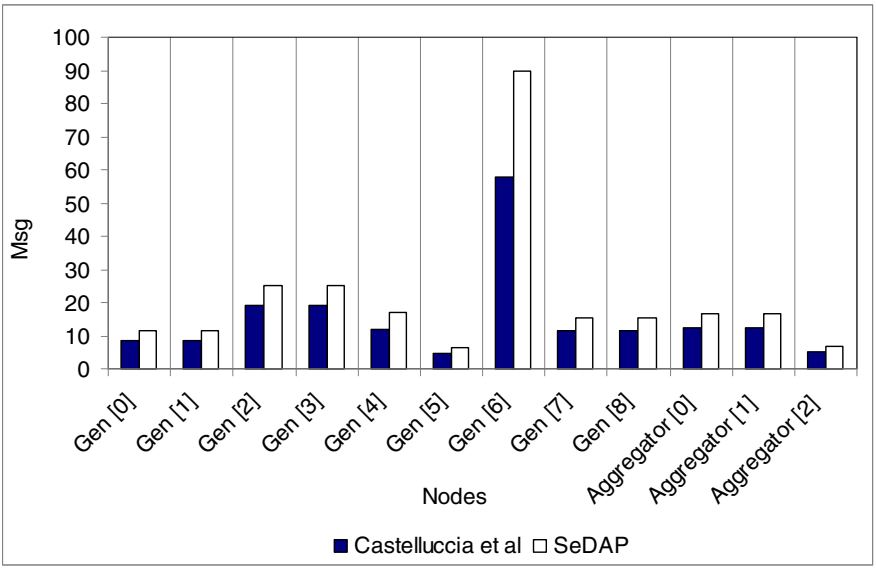

Fig. 3. Average No. of transmitted messages: SeDAP vs Castelluccia et al

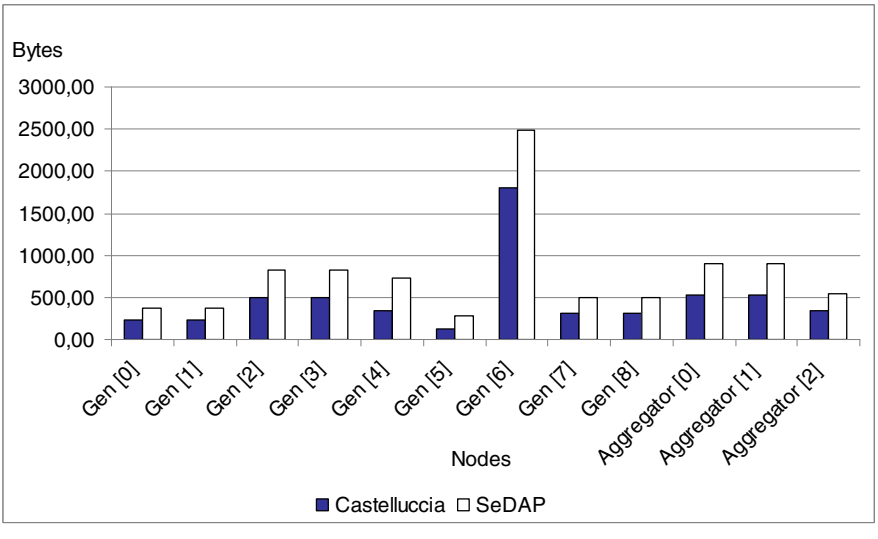

Fig. 4. Average No. of transmitted bytes: SeDAP vs Castelluccia et al

SeDAP transmits more messages and more bytes than a network that uses only the secure end-to-end aggregation [2]. However SeDAP is able to reveal malicious behaviour. In fact as shown in Figure5, which represents the total amount of received data from the sink , it is possible to notice that the 50% of the received data contains an error, but using SeDAP the sink knows the malicious behaviour thanks to the

notification error messages. Moreover, the delay between message that contains corrupted data and the related error notification message, sent by the controller, is shown in Figure 6. Summarizing the cost in terms of transmitted messages is balanced by the capability of SeDAP of revealing malicious behaviour, satisfying the anonymity and integrity requirement.

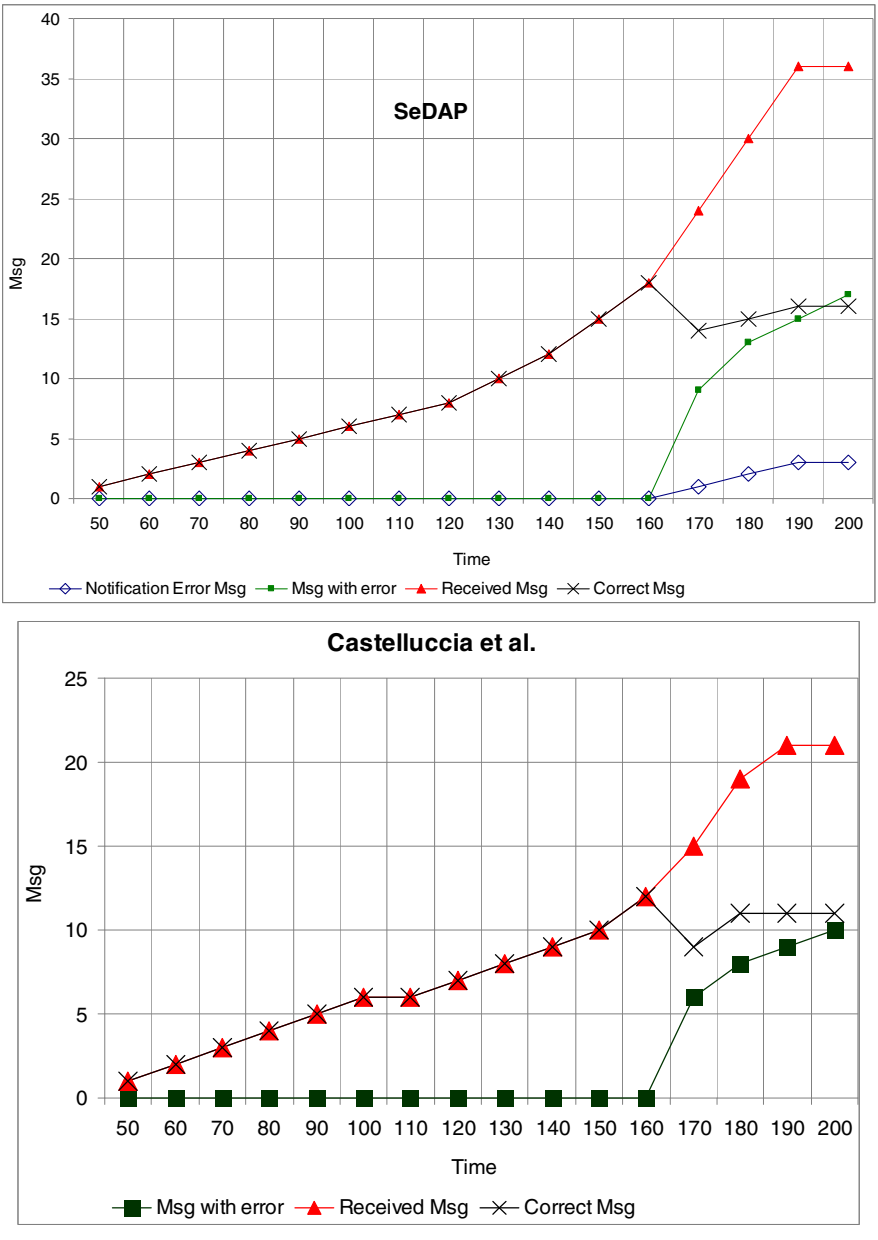

Fig. 5. No. of received messages: SeDAP vs Castelluccia et al

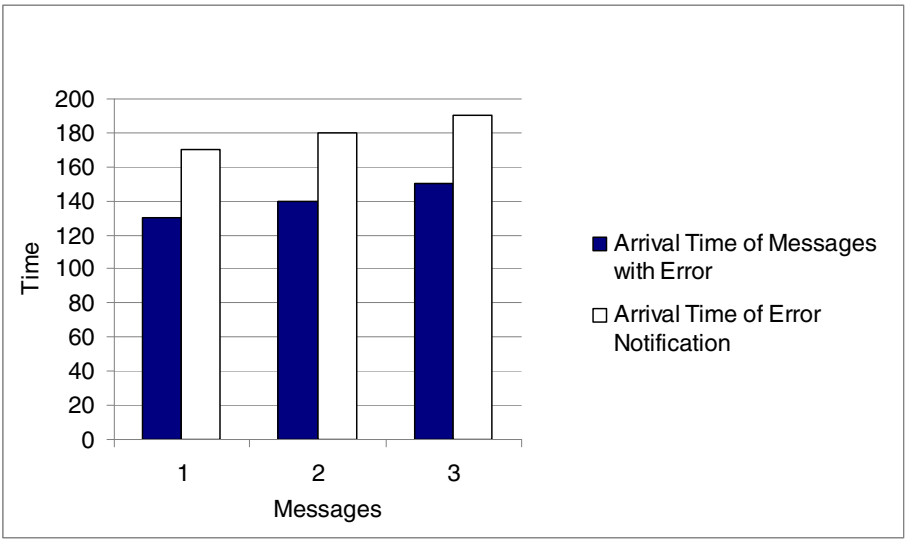

Fig. 6. Evaluation of delay between the arrival time of messages with errors and the arrival time of error notification messages

6 Related Works

WSN applications require to collect a huge amount of data and, due to the limited resources in terms of power of sensor nodes, it is necessary to aggregate such data in order to reduce the amount of transmitted information. Data may be used to perform attack towards privacy, integrity and confidentiality. Notice that the risk of violation increases due to both the wireless nature of the communication channel and the remote access. Exploiting such vulnerabilities the following common threats may occur [17], [18]:

- Eavesdropping: malicious users could easily discover the communication content listening to data.
- Masking: some malicious nodes may mask their real nature behind the identity of nodes that are authorized to take part to communication, and misroute the messages.

Designing secure WSN is a very mature research field [5] and the literature is very reach of solutions addressing at the same time aggregation issues and security aspects, such as confidentiality, integrity, authentication, and availability (an exhaustive and very comprehensive view of this topic can be found in [6]). Nevertheless, to the best of our knowledge, no solution is able to take into account both privacy and data aggregation issues at the same time using end-to-end encryption techniques.

As regards WSN privacy, the available solutions defined, starting from their vulnerabilities and related threats, may be classified into two main groups: anonymity mechanisms based on data cloaking [17] and privacy policy based approaches [19]. Data cloaking anonymity mechanisms perturb data following some kind of criterion,

for instance K-anonymity guarantees that every record is indistinguishable from at least k-1 other records [20].

In [17],[21],[22],[27], four main data cloaking anonymity approaches are proposed:

- Decentralize Sensible Data: the basic idea of this approach is to distribute the sensed location data through a spanning tree, so that no single node holds the complete view of the original data.
- Secure Communication Channel: the use of a secure communication protocols, such as SPINS [22], reduces the eavesdropping and active attack risk by means of encryption techniques.
- Change Data Traffic: the traffic pattern is altered with some bogus data that obfuscate the real position of the nodes.
- Node Mobility: the basic idea is to move the sensor nodes in order to change dynamically the localization information, making it difficult to identify the node.

For instance, [17] proposes a solution that guarantees the anonymous usage of location based information. More specifically, such a solution consists of a cloaking algorithm which regulates the granularity of location information to meet the specified anonymity constraints. This work only focuses on localization services and therefore, constrains the middleware architecture required to support the proposed algorithm. Hence, such a solution cannot be considered a general context independent anonymity approach.

Privacy policy based approaches [19], [24] state who can use individuals data, which data can be collected, for what purpose the data can be used, and how they can be distributed. A common policy based approach addresses privacy concerns at database layer after data have been collected [23]. Other works [24] address the access control and authentication issues, for instance Duri et al.[19] propose a policy based framework for protecting sensor information. Our work provides a contribution in the field of privacy policy based approaches by defining a role-based context-independent solution that guarantees anonymity of the nodes before sensed data are collected into a database. Our solution may be combined with both data cloaking mechanisms and some other privacy policy based approaches.

As regard secure data aggregation approaches proposed so far can be classified into two big families depending on whether the hop-by-hop or end-to-end cryptography is used. Hop-by-hop encryption is usually based on symmetric key schemes, which demand less computing resources than asymmetric key ones. These algorithms, such as [5], [6], [15], [25], [26], require that each aggregator must decrypt every message

it receives to enable in-network processing, thus causing a confidentiality breach. Furthermore, applying several consecutive encryption/decryption operations can negatively impair latencies. Finally, hop-by-hop aggregation requires each node to share secret keys with all its neighbours. To face these problems, literature has recently proposed aggregation algorithms able to work on ciphered data, using either asymmetric or symmetric keys [26]. The main limitations of such approaches being they allow very simple aggregation functions to be used, such as sum and average [6]. Despite this very broad variety of proposals, no single solution has been conceived yet to address confidentiality, integrity, anonymity and adaptive aggregation at the same time.

8 Conclusion

The present work proposed a protocol, SeDAP, that provides an integrated framework for facing the privacy and end-to-end secure data aggregation issues at the same time.

The definition of SeDAP has been supported by means of an ad-hoc UML conceptual model for the definition of privacy policies in the context of Wireless Sensor Networks. The model provides the basic concepts involved when dealing with the management of privacy-related information in a WSN. The efficiency of the proposed solution has been verified using simulations. Results show that SeDAP is able to guarantee node's anonymity and to identify malicious behaviour in a really short interval time.

At present we are extending SeDAP with a more powerful aggregation algorithm able to reduce the network load in case of congestion.

References

1. Akyildiz, I.F., Melodia, T., Chowdhury, K.: A survey on wireless multimedia sensor networks. Elsevier Computer Networks Journal (March 2007)
2. Castelluccia, C., Mykletun, E., Tsudik, G.: Efficient aggregation of encrypted data in wireless sensor networks. In: Conference on Mobile and Ubiquitous Systems: Networking and Services (2005)
3. Fasolo, Rossi, M., Widmer, J., Zorzi, M.: In-network aggregation techniques for wireless sensor networks: A survey. IEEE Wireless Communications (April 2007)
4. Mastrocristino, T., Tesoriere, G., Grieco, L.A., Boggia, G., Palattella, M.R., Camarda, P.: Control based on data-aggregation for wireless sensor networks. In: Proc. of IEEE Int. Symp. on Industrial Electronics, ISIE 2010, Bari, Italy (July 2010)
5. Grieco, L.A., Boggia, G., Sicari, S., Colombo, P.: Secure wireless multimedia sensor networks: a survey. In: Proc. of The Third Int. Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies, UBICOMM, Sliema, Malta (October 2009)
6. Ozdemir, S., Xiao, Y.: Secure data aggregation in wireless sensor networks: a comprehensive overview. Computer Networks 53 (2009)
7. Coen-Porisini, A., Colombo, P., Sicari, S., Trombetta, A.: A conceptual model for privacy policies. In: Proc. of SEA 2007, Cambridge (MS), USA (2007)
8. Coen-Porisini, A., Colombo, P., Sicari, S.: Dealing with anonymity in wireless sensor networks. In: Proceedings of 25th Annual ACM Symposium on Applied Computing (ACM SAC), Sierre, Switzerland (2010)
9. Coen-Porisini, A., Colombo, P., Sicari, S.: Privacy aware systems: from models to patterns. In: Mouratidis, H. (ed.) Software Engineering for Secure Systems: Industrial and Research Perspectives. IGI Global (2010)
10. Unified Modeling Language: Infrastructure, Ver. 2.1.2, OMG, formal/2007-11-02 (November 2007)
11. Unified Modeling Language: Superstructure, Ver. 2.1.2, OMG, formal/2007-11-02 (November 2007)
12. Directive 95/46/EC of the European Parliament. Official Journal of the European Communities, (L.281), 31 (November 23, 1995)

13. Ni, Q., Trombetta, A., Bertino, E., Lobo, J.: Privacy-aware role based access control. In: Proceedings of the 12th ACM Symposium on Access Control Models and Technologies (2007)
14. Zhanga, H., Arorab, A., Choic, Y., Goudac, M.: Reliable bursty convergecast in wireless sensor networks. Elsevier Computer Communications 30(13), 2560–2576 (2007)
15. Sicari, S., Riggio, R.: Secure aggregation in hybrid mesh/sensor networks. In: Proceeding of the IEEE International Workshop on Scalable Ad Hoc and Sensor Networks (SASN 2009), St. Petersburg, Russia, October 12-13 (2009)
16. OMNeT++ Discrete Event Simulation System (2005), http://www.omnetpp.org/doc/manual/usman.html
17. Gruteser, M., Schelle, G., Jain, A., Han, R., Grunwald, D.: Privacy-aware location sensor networks. In: Proceedings of the 9th USENIX Workshop on Hot Topics in Operating Systems, HotOS IX (2003)
18. Chan, H., Perrig, A.: Security and privacy in sensor networks. IEEE Computer Magazine, 103–105 (March 2003)
19. Duri, M.G.S., Liu, P.M.X., Perez, R., Singh, M., Tang, J.: Framework for security and privacy in automotive telematics. In: Proceedings of 2nd ACM International Worksphop on Mobile Commerce (2000)
20. Samarati, P., Sweeney, L.: Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical Report SRI-CSL-98-04, Computer Science Laboratory, SRI International (1998)
21. Gruteser, M., Grunwald, D.: A methodological assessment of location privacy risks in wireless hotspot networks. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, pp. 10–24. Springer, Heidelberg (2004)
22. Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E.: Spins: security protocols for sensor networks. Wireless Networking 8(5), 521–534 (2002)
23. Snekkenes, E.: Concepts for personal location privacy policies. In: Proceedings of 3rd ACM Conf. on Electronic Commerce (2001)
24. Molnar, D., Wagner, D.: Privacy and security in library rfid: Issues, practices, and architectures. In: Proceedings of ACM CCS (2004)
25. Yang, Y., Wang, X., Zhu, S., Cao, G.: Sdap: a secure hop-by-hop data aggregation protocol for sensor networks. In: ACM MOBIHOC (September 2006)
26. Westhoff, D., Girao, J., Acharya, M.: Concealed data aggregation for reverse multicast traffic in sensor networks: encryption key distribution and routing adaptation. IEEE Trans. Mobile Comput. 5, 1417–1431 (2006)
27. Smailagic, A., Siewiorek, D.P., Anhalt, J., Kogan, Y.W.D.: Location sensing and privacy in a context aware computing environment. In: Proceedings of Pervasive Computing (2001)