Computer Networks 56 (2012) 3655–3665

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Improving data quality using a cross layer protocol in wireless sensor networks

Alberto Coen-Porisini, Sabrina Sicari *

Dipartimento di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, Via Mazzini, 5, 21100 Varese, Italy

ARTICLE INFO

Article history:
Received 20 February 2012
Accepted 5 August 2012
Available online 17 August 2012

Keywords:
WSN
Privacy
Secure localization
Data quality
Simulation

ABSTRACT

Wireless sensor networks (WSNs) in which the location of sensors is a key information are becoming more and more important. Moreover, in many cases privacy is a key issues for such network which can be the target of different kind of security attacks. In this paper we present an approach, named *Cross Layer Protocol* (CLP), for improving data quality based on an integrated solution that considers a sound privacy management policy coupled with a secure localization protocol. More specifically, CLP exploits consistency between the information on nodes behavior gathered during localization phase and privacy compliance verification to evaluate nodes reputation. Finally CLP effectiveness is evaluated by means of a set of simulations.

1. Introduction

Wireless Sensor Networks (WSNs) 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.

In many applications contexts the location of sensor nodes [1] is an important information that can be used to identify the location of an event of interest (e.g., the location of an intruder, the location of a fire, etc.). In other cases, location information facilitates application services, such as location directory services (e.g., locating medical equipment and personnel in a smart hospital, locating survivors in debris, etc). Finally, it can be used in system functionalities such as geographical routing and location-based information querying. Hence, location-aware sensors are becoming the *de facto* standard in all application domains requiring location-based service.

The simplest way to provide localization information consists in equipping each sensor with a GPS receiver. However, this solution is not feasible from an economic nor a technical point of view since sensors are often deployed in large numbers and require manual configuration. A feasible approach, instead, consists in having nodes cooperating among them in order to compute their position. However, the main drawback is that several security attacks, such as node displacement, distance enlargement (by introducing fake nodes), dissemination of false position and distance information (by compromising nodes) can take place. Hence, trustworthiness and security of localization information are fundamental requirements in WSN.

Privacy is another crucial issue for many WSN applications such as localization and telemedicine. Moreover, in many other application contexts in which data referring to individuals are not directly handled by the WSN, privacy needs to be taken into account. For example, in home networks, sensor nodes may collect a large amount of data

* Corresponding author. Tel.: +39 0332 218 924; fax: +39 0332 218 919.
E-mail addresses: alberto.coenporisini@uninsubria.it (A. Coen-Porisini), sabrina.sicari@uninsubria.it (S. Sicari).

1389-1286/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.comnet.2012.08.001

that may reveal habits of individuals, violating in this way their privacy. However, wireless communications and the deployment in uncontrolled environments raise several issues in order to guarantee privacy (and security) since malicious tampering of sensors and/or traffic may jeopardize the confidentiality, the integrity, and the availability of data.

Traditional approaches to security and privacy are based on access control and strong authentication. Unfortunately neither of them are suitable to WSN because of the limited resources and short battery life of sensors. Moreover, approaches based on pre-shared encryption keys are prone to physical attacks since sensor devices and their key can be easily cloned.

This paper, which extends the preliminary results presented in [24], tackles both secure localization and privacy issues in order to define an integrated solution that considers a sound privacy management policy coupled with a secure localization protocol. The presented approach is based on the assessment of data quality, that is we evaluate to which extent the information to be processed by applications is reliable and trustworthy. This is done by introducing a way to evaluate the overall data quality when several cheap protection techniques are combined together. Although none of the used techniques guarantees reliability and trustworthy by itself, we exploit consistency across them to evaluate data reliability. As a result we introduce a protocol, named Cross-Layer Protocol (CLP), that defines the fundamental steps for assessing data quality.

The rest of the paper is organized as follows: Section 2 introduces the foundations for modeling privacy and describes the reference scenario. Section 3 introduces CLP that integrates privacy management policies and secure localization for cross-layer data assessment. Section 4 reports a set of simulations whose aim is to evaluate CLP. Section 5 presents some related works. Finally, Section 6 draws some conclusions and provides hints for future works.

2. Foundations

In what follows we provide a short description of the privacy model used in this paper, which is discussed more in detail in [2–4], along with the assumptions made on both the network and the messages.

2.1. 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 [7]. Notice that, since the proposed terms are general, i.e., 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

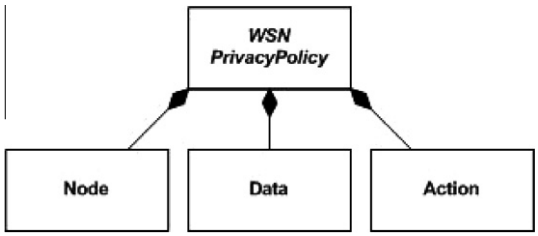

Fig. 1. A WSN privacy policy.

and their relationships. Fig. 1 depicts a class diagram that provides a high level view of the basic structural elements of the model.

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.

Node represents a member of the network and it is characterized by a function and a role. The former describes the task performed by the node within the network in which it operates (e.g., data sensing, message transmission, message forwarding, data aggregation, etc.), while the latter describes the role played by the node 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.

Data represents the information handled by processors and is extended by *Identifiable* data and *Sensed* data. The former represents the information that can be used to uniquely identify nodes (e.g., node identifier) while the latter represents the information that is sensed by the nodes of the network (e.g., temperature, pressure). Moreover, *Sensed* data is further extended by means of *Sensitive* data, which represents the information that deserves particular care and that should not be freely accessible (e.g., health related data). *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. Moreover, data are aggregated among them to compose Messages, which represents the basic communication unit exchanged by the nodes of the network.

Action represents any operation performed by *Node* and 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 in the context of WSN each function usually corresponds to one action.

In order 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 encryption

mechanisms are used. More specifically, two classes representing encryption keys, named *DataKey* and *FunctionRoleKey*, are introduced. The former key is used to protect sensed data; while the latter is used to ensure that message communication and data handling are executed only by authorized nodes.

2.2. The network

We consider a dense network composed of N nodes in which each node senses a given type of data (e.g., temperature, pressure, brightness, position, and so on). Nodes are uniquely identified by means of a label n and can exchange messages so that all sensed data are directed to the sink. Each node directly communicates with its closer neighbors (at one hop distance) and thus, a sensed data before reaching the sink passes through different nodes of the network (multi-hop communication) by means of different messages. Notice that 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 [9].

Messages represent a single transmission hop between adjacent nodes and contain data that may be classified as *identifiable* and *sensed*. A message is denoted by $msg_{n,q}$, where n identifies the node that generated and transmitted the message and q identifies the message among those generated by node n. Thus, the pair (n,q) unambiguously identifies the message among those transmitted in the network. In order to guarantee the integrity and confidentiality of the end-to-end communication, we use a message structure that keeps track of the last two hops of the transmission. Therefore, a message $msg_{n,q}$, is a tuple.

$msg_{n,q} = \langle curr, prev, sub, sensId, errId, errFlag, data, idList \rangle$
where:

- *curr*: is the couple $\langle n,q \rangle$, which unambiguously identifies the current message among those transmitted within the network by node n.
- *prev*: is a couple $\langle n_p, q_p \rangle$, where n_p, is the identifier of the node that operated the second last forwarding of the sensed data contained in the current message, and q_p is the identifier used by n_p to identify such a message.
- *sub*: is a couple $\langle n_s, q_s \rangle$ where n_s is the identifier of the node that originally sensed the data, and q_s is the message identifier used by such a node 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 found the error.
- *sensId*: is a couple $\langle n_{si}, q_{si} \rangle$ that in case of error notification contains the identifier of the node that sensed the correct data and the identifier of the message transmitted by such a node.
- *errId*: is a tuple $\langle n_{ei}, q_{ei} \rangle$, 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.
- *errFlag*: represents an error code reporting whether an anomaly was identified in the message content.
- *data*: includes the ciphered data c either sensed or aggregated by the subject node or the aggregator node, respectively.
- *idList:* is a list containing the identifiers of the nodes that already processed the data content of the message.

Notice that fields *sensId* and *errId* are used only when *errFlag* equals 1, that is the message reports an error notification.

In order to guarantee the confidentiality of messages content every field but *errFlag* is ciphered. Notice that a node may play different functions and roles and therefore it may own multiple function-role keys (one for each pair of function-role). More specifically the following function-role pairs are defined: Sensing-Subject (SS), Authenticator–Processor (AP), Transmitter–Processor (TP) and Notifier–Controller (NC). 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.[1] 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.

At sink level, nodes are classified, as far as localization is concerned, in *Verifier* and *Unknown* nodes. The former are nodes whose position is known, while the latter are those whose position is unknown. Notice that Verifier nodes are able to cooperate among them to verify the position of an unknown node.

3. CLP: the proposed solution

In what follows the protocols introduced in order to guarantee secure localization and to meet privacy requirements are presented. More specifically the protocols introduced are the following:

- *Sensing*, which defines the actions that a node carries out to communicate sensed data to the other nodes of the network
- *Message Reception and Integrity Verification*, which defines the actions that a node carries out when receiving a message from other nodes in order to verify its integrity and possibly to re-transmit it over the network
- *Secure localization*, which defines the action that a node carries out to localize in secure manner. The output of this phase are the coordinates of the node along with an evaluation of their trustworthiness, which are stored by the sink
- *Cross-layer node evaluation*, which defines the actions performed by the sink in order to evaluate nodes reputation using the information gathered from the localization phase and by evaluating privacy policies compliance.

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

The aim of the presented protocol is to guarantee (i) data integrity; (ii) anonymity and (iii) secure node localization by using cross-layer data evaluation.

3.1. The sensing protocol

Let n be a node sensing a data d from the environment where it is located. According to the function-role classification, when sensing d the node acts as a Sensing-Subject (SS) and therefore d is encrypted using key k (n,SS). Moreover, let q denote the number of messages that n already transmitted over the network. Thus, message $m_{n,q+1}$ is prepared according to the structure discussed in the previous section. Notice that, when preparing the message the node acts as a *Transmitter–Processor* (*TP*) and therefore every ciphered field but *data* is encrypted using key k (n,TP).

$msg_{n,q+1} = \langle curr, prev, sub, sensId, errId, errFlag, data, idList\rangle$, where
–$curr = \langle Enc(n, k(n,TP)),\ Enc(q+1, k(n,TP))\rangle$;
–$prev = \varepsilon$[a];
–$sub = \langle Enc(n, k(n,TP)),\ Enc(q+1, k(n,TP))\rangle$;
–$sendId = \varepsilon$;
–$errId = \varepsilon$;
–$errFlag = 0$;
–$data = Enc(d, k(n,SS))$;
–$idList = \{Enc(n, k(n,TP))\}$.

[a] ε represents an empty field.

Once prepared part of the message (fields *data* and *subject*) is stored in the local table before being put in the transmission queue.

3.2. Message reception and integrity verification protocol

Let n be a node receiving a message $m_{j,h}$, and let q be the number of messages already transmitted by n over the network. The message is analyzed to find out whether it was originally transmitted by the node itself. This can be done searching the local table using the content of field *prev* as hash key. If the search fails n has to re-transmit the message over the network, that is n acts as a *Transmitter–Processor* (*TP*). Thus a new message $msg_{n,q+1}$ is prepared and then it is stored in the local table before being put in transmission queue.

$msg_{n,q+1} = \langle curr, prev, sub, sensId, errId, errFlag, data, idList\rangle$, where
–$curr = \langle Enc(n, k(n,TP)),\ Enc(q+1, k(n,TP))\rangle$;
–$prev = m_{j,h}.curr$[a];
–$sub = m_{j,h}.sub$;
–$sendId = m_{j,h}.sendId$;
–$errId = \varepsilon$;
–$errFlag = 0$;
–$data = m_{j,h}.data$;
–$idList = m_{j,h}.idList \cup \{Enc(n, k(n,TP))\}$.

[a] $m_{j,h}$.curr stands for field curr of message $m_{j,h}$

Instead if the search succeeds, then $m_{j,h}$ was transmitted by n and therefore the integrity of the received message is verified, that is n acts as a *Notifier–Controller* (*NC*). Hence, the node compares the content of field *data*, of the received message with the information retrieved from its table. If the information matches, 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 one 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 $msg_{n,q+1}$ in order to notify the sink that a corrupted message is spreading through the network.

$msg_{n,q+1} = \langle curr, prev, sub, sensId, errId, errFlag, data, idList\rangle$, where
–$curr = \langle Enc(n, k(n,TP)),\ Enc(q+1, k(n,TP))\rangle$;
–$prev = \varepsilon$;
–$sub = \langle Enc(n, k(n,TP)),\ Enc(q+1, k(n,TP))\rangle$;
–*sendId = retrieved.sub;*
–$errId = m_{j,h}.curr$;
–$errFlag = 1$;
–$data = Enc(retrieved.data,\ k(n,NC))$;
–$idList = m_{j,h}.idList \cup \{Enc(n, k(n,TP))\}$.

Notice that *errFlag* is set to 1 to indicate that the current message is an error message; field *prev* is empty to avoid message loops with the malicious node and the spreading of error messages; both fields *sub* and *curr* are set to n to specify which node found the error; *sendId* equals field *sub* of the message stored in the local table, to report which node has sensed the original data that was found to be corrupted; *errId* equals field *curr* of the received message to report which node made the mistake. Finally, field *data* is set by encrypting with the *Notifier–Controller* key the homonymous field retrieved from the local table.

Once generated the message is stored in the local table before being put in the transmission queue.

3.3. Secure localization

Node positions are evaluated using a multilateration technique, which determines the node coordinates by exploiting a set of landmark nodes, called anchor nodes, whose 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 get a reply to a beacon message sent to u. This is done under the assumption that the speed of the signal in the medium in which the transmission occurs is known.

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 [22], which uses three or more anchor nodes to detect misbehaving nodes. In what follows we briefly report the VM technique for the sake of completeness (see Fig. 2).

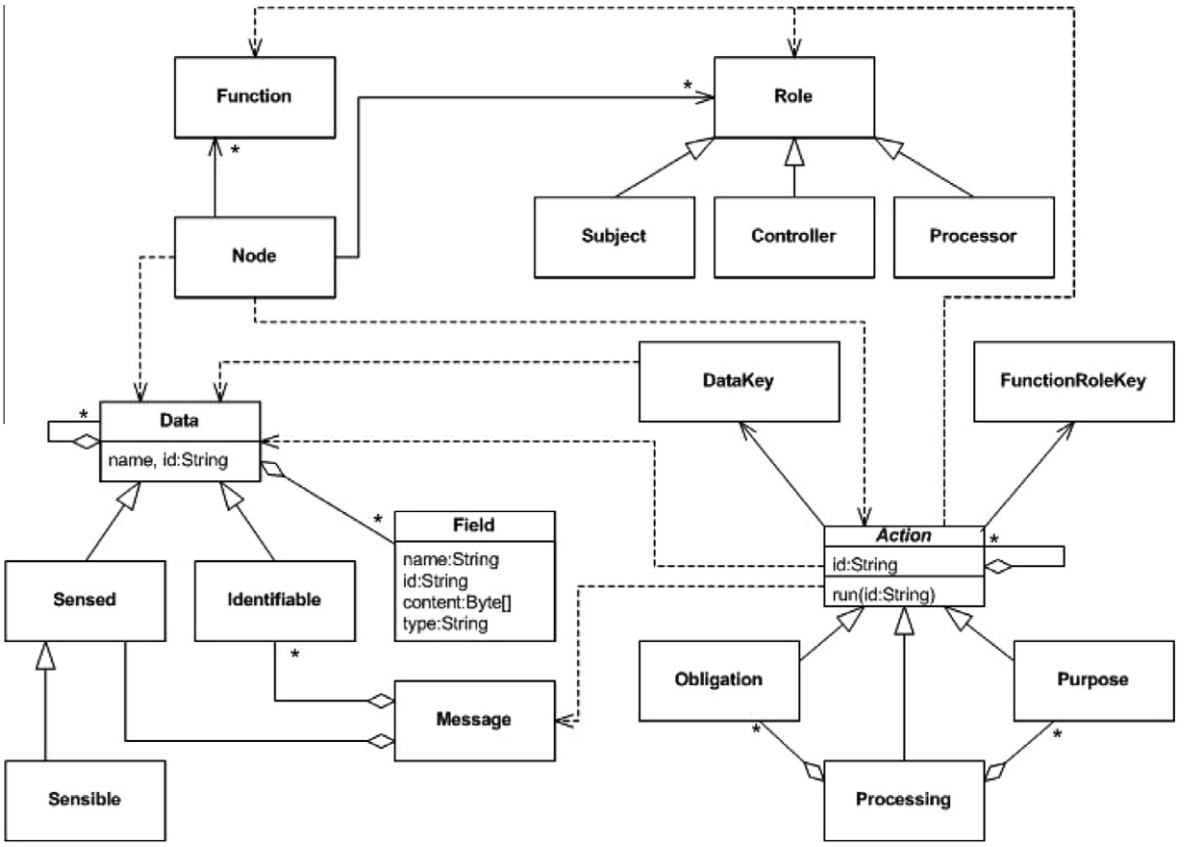

Fig. 2. UML model.

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. Since this cannot be done it is possible to figure out that u is misbehaving. As an example Fig. 3 shows a node u pretending to be in a position farther away from V_1 than the actual position. As a consequence the position of u would result closer to V_3.

Therefore, if the verifiers are trusted and they can communicate securely the sink can check the localization data, in the following way. Let T_1, T_2 and T_3 be the time needed to get an answer from u to the beacon message sent by V_1, V_2 and V_3, respectively. Starting from T_i the corresponding distance bound db_i [23] is computed,[2] for $1 \leqslant i \leqslant 3$.

Let $\langle x_u, y_u \rangle$ denote the coordinates of the estimated position of u, and let $f_i(x_u, y_u)$ denote the function representing the difference between the distance bound and the estimated distance of u from V_i.

$$f_i(x_u, y_u) = db_i - \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2} \tag{1}$$

[2] The distance bound db_i is the upper bound of the distance between V_i and u.

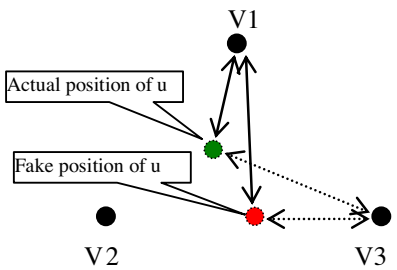

Fig. 3. Verifiable multilateration.

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

$$F(x_u, y_u) = \sum_{i=1}^{3} (f_i(x_u, y_u))^2 \tag{2}$$

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, $\langle x_u, y_u \rangle$, is considered correct if $f_i(x_u, y_u) < \delta_{err}$, for $1 \leqslant i \leqslant 3$. 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 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 sink considers the estimated position as correct and therefore node u is marked as *Robust*.

3.4. Cross-layer node evaluation

The sink evaluates the trustworthiness of the nodes of the network by looking at the messages it receives and by using the information gathered during the localization phase. Notice that the sink uses a node reputation table to store information about nodes trustworthiness. Such a table reports for each node two different values, the first of which provides information about node localization (i.e., *Robust*, *Malicious* or *Unknown*), while the second one provides information about privacy compliance (i.e., *PrivacyCompliant* or *PrivacyViolation*). Notice that initially, anchor nodes (i.e., verifiers) are considered to be *Robust*, while the remaining nodes are classified as *Unknown*. Moreover, initially all nodes are considered to be *PrivacyCompliant*.

Each time the sink receives a message it carries out the evaluation by checking whether field *errFlag* is set to 1 or not. If it is, this means that the received message is an error notification message. As a consequence, the reputation of the node whose identifier is reported by field *idErr* (i.e., the node that made the mistakes reported by the message) is updated by assigning the value *PrivacyViolation*. Notice that, in such a case the field *data* of the message contains the correct message, which can be further processed by the sink.

Otherwise, if field *errFlag* equals 0 then the received message contains sensed data and therefore the sink before processing data evaluates the trustworthiness of all the nodes that handled the sensed data (i.e., the nodes whose identifiers are stored in fields *sub*, *idList* and *curr*). This is done by verifying their reputation as stored in the reputation table.

If the reputation is *Robust* and *PrivacyCompliant* the sink considers the data as reliable; otherwise if the reputation is *Malicious* or *PrivacyViolation* the data are discarded; finally if the reputation is *Unknown* and *PrivacyCompliant* the data may be processed or discarded depending on the sink policy. For example, let us consider a node n whose reputation is *Unknown*, in such a case the sink may take into account the reputation of the neighboring nodes (i.e., nodes that can communicate directly with n) and discard the data if any of them is classified as *Malicious*.

Finally, it must be noticed that a malicious node may decide not to lie on its position, still providing fake information in term of sensed data. In order to uncover this kind of malicious behaviors other consistency properties can be exploited. For example, the sink of a system monitoring temperature may receive from a node a datum of 40 °C, while all the others nodes report 20 °C. Since such an abrupt change in temperature may be considered as anomalous, the sink may decide to discard the former information. However, this kind of data validation depends highly on the application domain and therefore no general policy can be provided.

Notice that even if fake data may be produced by a node that provided authentic localization information, knowing the real position of the malicious node may help the sink to take appropriate counter-measures. In conclusion, cross-layer analysis enables a more careful assessment of the overall quality of the received data, thus avoiding malicious poisoning.

4. Performance evaluation

In order to evaluate the effectiveness of the solution presented in this paper we simulated the behavior of a wireless sensor network measuring the temperature of a given environment. The aim of the simulations was to compare the behavior of a network that uses both secure localization and privacy compliance with the behavior of a network that adopts either secure localization, such as Capkun et al. [22] or a privacy aware solution such as Coen et al. [3].

The simulations were carried out using Omnet++ [10] and aimed at evaluating the presented approach with

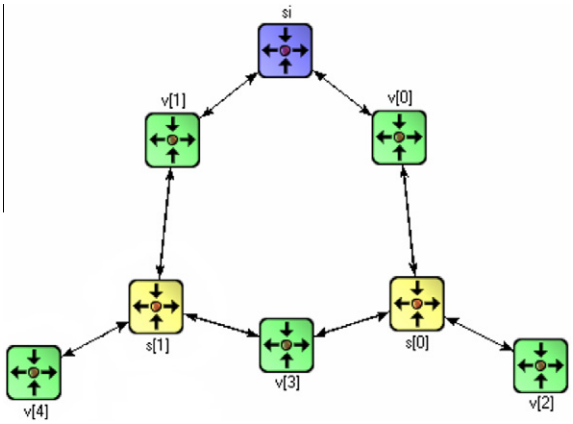

Fig. 4. The first network topology.

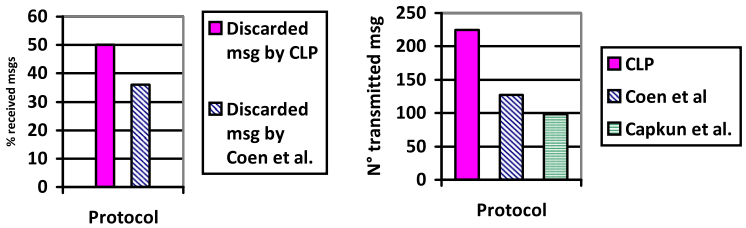

Fig. 5. % Of discarded messages and no. of transmitted messages.

respect to different aspects such as the capability to identify malicious nodes, the robustness to an increasing number of malicious nodes and the robustness to different network topologies.

The first network taken into account is reported in Fig. 4 and is composed by five verifiers $v[0 \cdots 4]$, two sensors $s[0 \cdots 1]$, and the sink *si*. Moreover, we assume that only node $s[1]$ is behaving maliciously, that is it provides fake localization information and it violates the privacy policy by tampering messages.

The results, reported in Fig. 5 show that by combining localization and privacy, it is possible to identify 14% more of corrupted messages in comparison with Coen et al., while the number of messages transmitted over the network increased by 44% with respect to Coen et al. and by 57% with respect to Capkun et al.

Notice that node $s[1]$ in order to communicate a fake position needs to forge localization information sent by verifier $v[4]$. This can be done since messages coming from verifier $v[4]$ go through $s[1]$ in order to reach the sink. In other words $s[1]$ can isolate $v[4]$ from the rest of the network. Thus, only localization may not be enough to detect the malicious behavior of $s[1]$. However, if $s[1]$ modifies messages coming from $v[4]$, most likely it will modify also messages coming from other nodes and this would cause notification error messages to be sent to the sink. As a consequence the sink will downgrade the reputation of $s[1]$ to *PrivacyViolation*.

In general in order to prevent attacks based on isolation of anchor nodes it is necessary that their position in the network is such that at least two different paths that can be used to send messages to the sink exist.

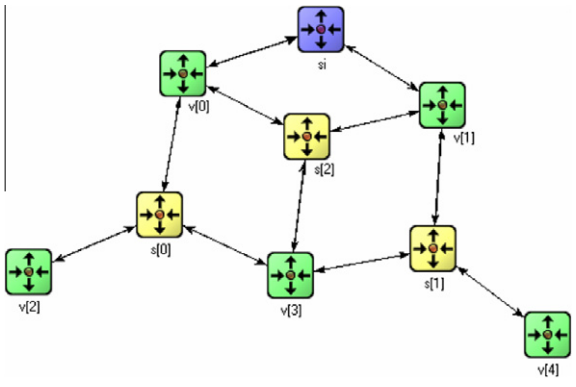

Fig. 6. The second network topology.

As a second example we considered a network, reported in Fig. 6, comprising five verifiers $v[0 \cdots 4]$, three sensors $s[0 \cdots 2]$ and the sink *si*. Notice that we carried out several simulations to compare the results of our approach when either nodes $s[1]$ or $s[2]$ behaves maliciously with the case in which both of them do.

Fig. 7 reports the results that show that the number of discarded messages increases by 42% when node $s[2]$ behaves maliciously in addition to $s[1]$ and that the number of discarded messages depends on the position of the corrupted node.

In fact the network performance depends on the number of uncorrupted paths towards the sink. Thus, reducing the number of uncorrupted paths causes an increase in the number of discarded messages.

The above results show that the most important parameter is the position of the malicious node(s) rather than their number. In fact, the number of discarded messages increases with the number of communication paths in which a node behaves maliciously.

Fig. 8 shows another network that differs from the previous one in the number of verifiers (increased to seven) and in the density[3] of nodes distribution (the former is denser than the latter). Notice that this network has less paths to the sink with respect to the one of Fig. 6.

Fig. 9 reports the comparison between the two topologies, in term of number of discarded messages, assuming that node $s[1]$ behaves maliciously in both networks.

The results show that in a more dense network even if with less nodes the sink discards more messages.

Finally, in order to evaluate the behavior of CLP in a real setting we considered the network of Fig. 10, which is composed of 127 nodes (59 verifiers and 68 plain nodes).

This network was analyzed considering different kinds of malicious behaviors consisting in messages tampering (privacy violation) and/or fake location information. The first issue that we investigated was the relationship between the number of malicious nodes and the total number of messages transmitted over the network, since such a number can increase because of error notification messages that are transmitted towards the sink. The results, reported in Fig. 11, show that the number of messages

[3] In this context network density measures the presence of redundant paths towards the sink.

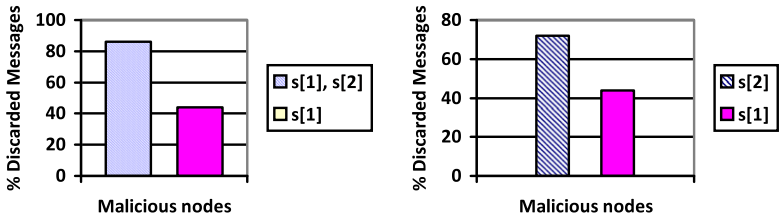

Fig. 7. Different no. of malicious nodes/different positions of the malicious node.

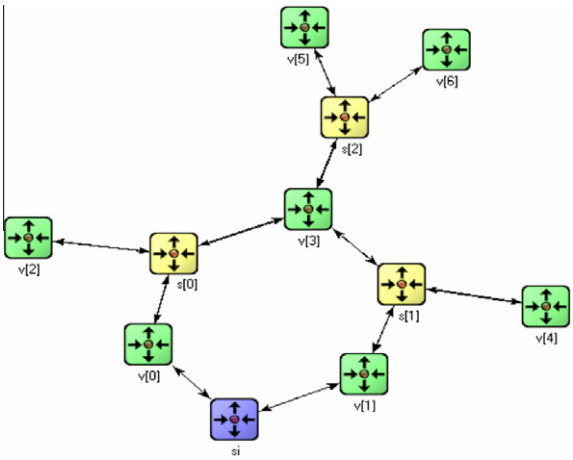

Fig. 8. The third network topology.

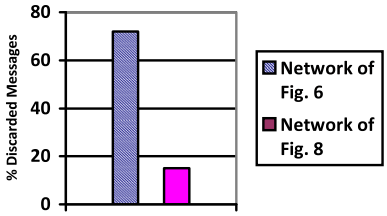

Fig. 9. Discarded messages in different topologies.

increases linearly with the number of malicious nodes in the network.

The second issue concerns whether CLP is able to identify all the malicious nodes in the network. For this purpose we set up two different scenarios, made of five malicious nodes. In the first one three nodes provide fake localization information, while the remaining two violate the privacy policy. In the second scenario one node provides fake localization information, two violate the privacy policy and the two present both misbehaviors.

The results, expressed in term of node classification, are reported in Figs. 12 and 13, respectively. Notice that the number of node classified as Robust decreases when using CLP with respect to the case in which only localization information is used (VM). Moreover, in both scenario CLP correctly identifies all the malicious nodes, while in the second scenario one of the nodes providing fake localization information is not detected by using only VM.

Thus, we decided to further investigate this aspect considering two more scenarios representing a stress test characterized by 25 malicious nodes, in which five nodes provide fake localization information, 15 violate the privacy policy and the remaining five do both violations. The difference between the two scenarios is the geographical distribution of the malicious nodes in the network.

The results, reported in Figs. 14 and 15, show that CLP identifies in both scenarios every malicious node in the network. In both cases we compared the results of CLP with those obtained by using the two techniques (VM

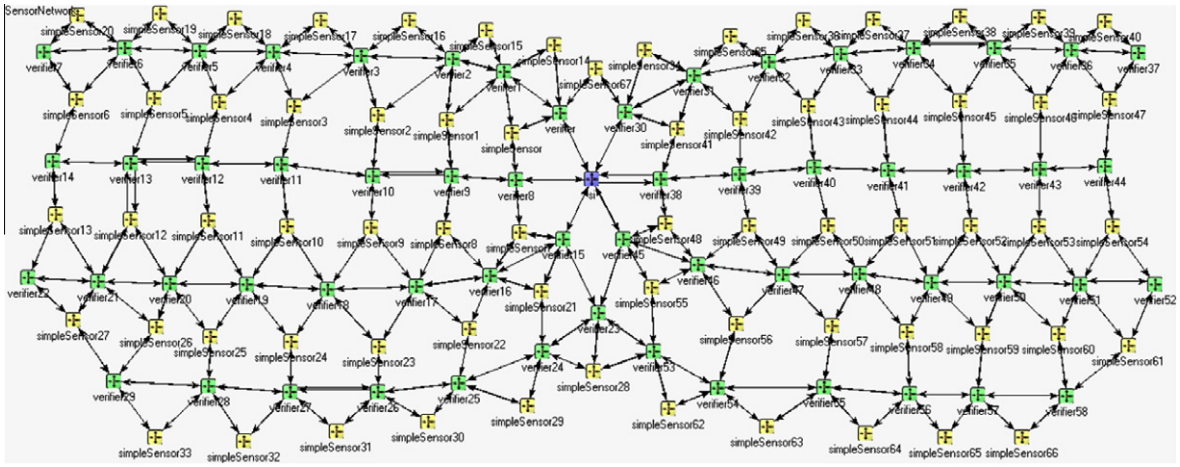

Fig. 10. The fourth network topology.

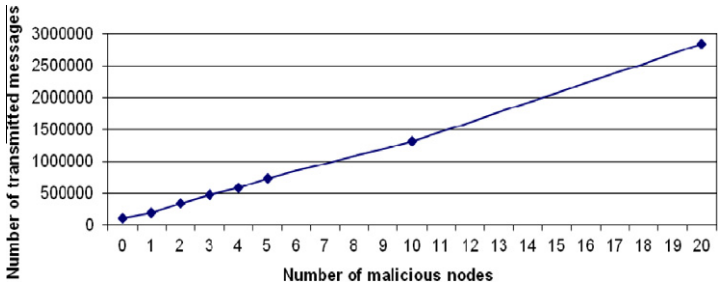

Fig. 11. No. of transmitted messages vs no. of malicious nodes.

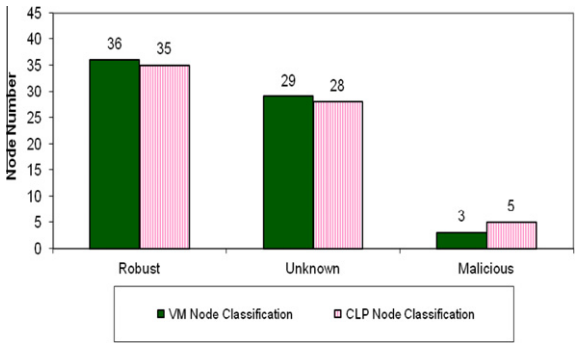

Fig. 12. Node classification VM vs CLP: three localization malicious and two privacy malicious.

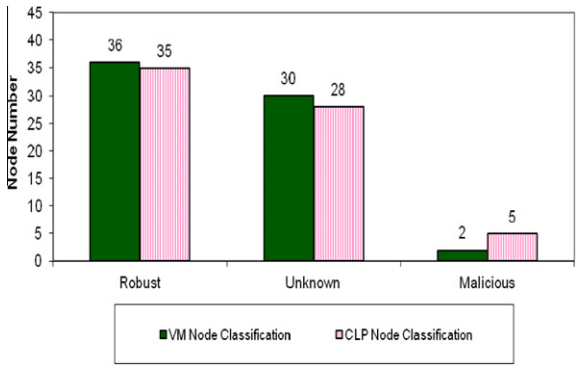

Fig. 13. Node classification VM vs CLP: 1 localization malicious, 2 privacy malicious and 2 malicious for both privacy and localization.

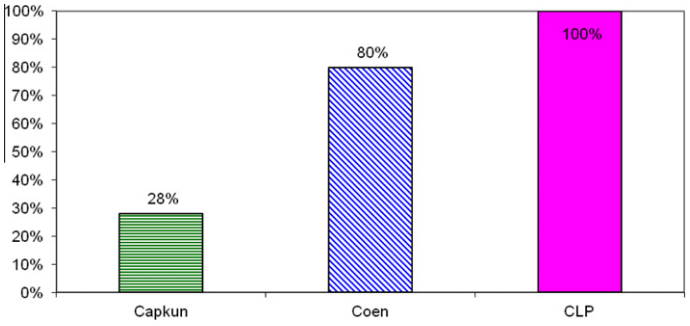

Fig. 14. Performance evaluation of CLP: identification of 25 malicious nodes.

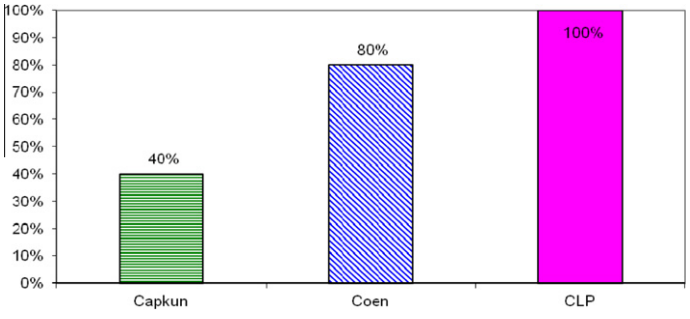

Fig. 15. Performance evaluation of CLP: identification of 25 malicious nodes with an increment of malicious node in localization phase.

and privacy violation) in an independent way. However, the result of Fig. 14 shows that in the first scenario, using only VM only seven out of 10 nodes providing fake localization information are detected,[4] that is three nodes are not identified as malicious due to the lack of information provided by the verifiers.

Finally, we carried out some more simulations to see what kind of relation exists among the number of messages transmitted over the network and other parameters such as the position of the malicious nodes and the number of verifiers in the network. In both cases the results showed that such parameters have not a significant influence on the number of transmitted messages, while they can affect the effectiveness of the approach.

5. Related works

In WSN, the wireless nature of the communication channel and the remote access increase the risk of privacy, integrity and confidentiality violations. In fact, exploiting such intrinsic vulnerabilities the following common threats may occur [12,13]:

- *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 and the literature reports many 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 [12]). Nevertheless, to the best of our knowledge, no solution is able to take into account privacy, data integrity and secure localization issues at the same time using end-to-end encryption techniques.

As far as privacy is concerned, the available solutions may be classified into two main groups: anonymity mechanisms based on data cloaking [11] and privacy policy based approaches [17].

For instance, [11] 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.

Other approaches belonging to the former solution are K-Anonymity, which guarantees that every record is indistinguishable from at least $k-1$ other records [14]; Decentralize Sensible Data, in which sensed location data is distributed through a spanning tree, so that no single node holds the complete view of the original data; Secure Communication Channel, in which the use of a secure communication protocols, such as SPINS [16], reduces the eavesdropping and active attack risk by means of encryption techniques; Change Data Traffic, in which the traffic pattern is altered with some bogus data that obfuscate the real position of the nodes; Node Mobility, in which the sensor nodes are moved in order to change dynamically the localization information, making it difficult to identify the node.

Privacy policy based approaches [3,15,17,18] 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 [17]. Other works [18] address the access control and authentication issues, for instance Duri et al. [13] 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 and data integrity 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 far as data integrity is concerned, most of the proposed solutions are based on the adoption of encryption techniques, ad hoc key distribution schemes [19–21], authentication, access control solutions. Instead, our approach proposes to combine some cheap protection techniques even if none of them is totally effective. We rely on cross-layer evaluation to assess the overall quality of the collected information. Thus, by combining verification of localization information and the identification of privacy violations we evaluate nodes reputation and therefore data quality, without wasting power with some complex security countermeasures such as public key encryption techniques.

6. Conclusions

Data quality is a fundamental requirement in any WSN scenario. Although it is very difficult to provide data trustworthiness due to the distributed nature and the limited resource in terms of power, our approach allows the sink to analyze data trustworthiness by exploiting consistency on cross-layer information, i.e., node localization and privacy violations.

More specifically, the trustworthiness about the node position information and the privacy compliance are used for evaluating data trustworthiness. In fact node position, being target of different kind of attacks (e.g., malicious node displacement, distance enlargement) can be used to identify malicious behavior.

Our approach is largely independent from the adopted routing protocols, the verification localization algorithm and the used encryption technique. Besides assessing data trustworthiness we provide an integrated framework for

[4] The total number of nodes providing fake information localization is 10, i.e., 40% of the malicious nodes.

facing privacy and secure localization issues at the same time.

The effectiveness of the proposed solution has been tested by means of many simulations that showed that CLP, besides guaranteeing anonymity, provides secure node localization and the capability to identify malicious behaviors. Moreover the obtained results provided interesting hints on how networks should be designed and where anchor nodes should be placed.

References

[1] I.F. Akyildiz, T. Melodia, K. Chowdhury, A survey on wireless multimedia sensor networks, Elsevier Computer Networks Journal (2007).

[2] A. Coen-Porisini, P. Colombo, S. Sicari, A. Trombetta, A conceptual model for privacy policies, in: Proceedings of SEA 2007, Cambridge (MS), USA.

[3] A.Coen-Porisini, P.Colombo, S. Sicari, Dealing with anonymity in wireless sensor networks, in: Proceedings of 25th Annual ACM Symposium on Applied Computing (ACM SAC), Sierre, Switzerland, 2010.

[4] A. Coen-Porisini, P. Colombo, S. Sicari, Privacy aware systems: from models to patterns, in: H. Mouratidis (Ed.), Software Engineering for Secure Systems: Industrial and Research Perspectives, IGI Global, 2010.

[5] Unified Modeling Language: Infrastructure, Ver. 2.1.2, OMG, November 2007, formal/2007-11-02.

[6] Unified Modeling Language: Superstructure, Ver. 2.1.2, OMG, November 2007, formal/2007-11-02.

[7] Directive 95/46/EC of the European Parliament. Official Journal of the European Communities of 23 November 1995 No L. 281 p. 31.

[8] Q. Ni, A. Trombetta, E. Bertino, J. Lobo, Privacy-aware role based access control, in: Proceedings of the 12th ACM Symposium on Access Control Models and Technologies, 2007.

[9] H. Zhanga, A. Arorab, Y. Choic, M. Goudac, Reliable bursty convergecast in wireless sensor networks, Elsevier Computer Communications 30 (13) (2007) 2560–2576.

[10] OMNeT++ Discrete Event Simulation System <http://www.omnetpp.org/doc/manual/usman.html>, 2005.

[11] M. Gruteser, G. Schelle, A. Jain, R. Han, D. Grunwald, Privacy-aware location sensor networks, in: Proceedings of the 9th USENIX Workshop on Hot Topics in Operating Systems (HotOS IX), 2003.

[12] H. Chan, A. Perrig, Security and privacy in sensor networks, IEEE Computer Magazine (2003) 103–105.

[13] M.G.S. Duri, P.M.X. Liu, R. Perez, M. Singh, J. Tang, Framework for security and privacy in automotive telematics," in: Proceedings of 2nd ACM International Worksphop on Mobile Commerce, 2000.

[14] P. Samarati, L. Sweeney, 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.

[15] M. Gruteser, D. Grunwald, A methodological assessment of location privacy risks in wireless hotspot networks, in: Proceedings of the First International Conference on Security in Pervasive Computing, 2003.

[16] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, D.E. Culler, Spins: security protocols for sensor networks, Wireless Networking 8 (5) (2002) 521–534.

[17] E. Snekkenes, Concepts for personal location privacy policies, in: Proceedings of 3rd ACM Conference on Electronic Commerce, 2001.

[18] D. Molnar, D. Wagner, Privacy and security in library rfid: issues, practices, and architectures, in: Proceedings of ACM CCS, 2004.

[19] L. Eschenauer, V.D. Gligor, A key-management scheme for distributed sensor networks, in: Proceedings of 9th ACM Conference on Computer and Communications Security, 2002.

[20] R.D. Pietro, A. Mei, L.V. Mancini, Random key assignment for secure wireless sensor networks, in: Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASNs), Fairfax-VA, USA, 2003.

[21] R.D. Pietro, C. Soriente, A. Spognardi, G. Tsudik, Collaborative authentication in unattended wsns, in: Proceedings of 2nd ACM Conference on Wireless Network Security (WiSec), Zurich, Switzerland, 2009.

[22] S. Capkun, J. Hubaux, Secure positioning in wireless networks, in: IEEE Journal on Selected Areas in Communications, vol. 24(2), February 2006, pp. 221–232.

[23] S. Brands, D. Chaum, Distance-bounding protocols, in: Proceedings of Workshop on the Theory and Application of Cryptographic Techniques, 1994.

[24] A. Coen-Porisini, S.Sicari, Cross layer data assessment in wireless sensor networks, in: Proceeding of Sensornets 2012, International Conference on Sensor Networks, Rome, Italy, 24–26 February 2012.

Alberto Coen Porisini received his Dr. Eng. degree and Ph.D in Computer Engineering from Politecnico di Milano (Italy) in 1987 and 1992, respectively. He is Professor of Software Engineering at Università degli Studi dell'Insubria (Italy) since 2001 and Dean of the School of Science since 2006. Prior to that he was Associated Professor at Università degli Studi di Lecce (1998–2001), Assistant Professor at Politecnico di Milano (1993–2001) and Visiting Researcher with the Computer Security Group at University of California, Santa Barbara (1992–1993).
His main research interests are in the field of specification and design of real-time systems, privacy models and wireless sensor networks.

Sabrina Sicari is Assistant Professor at Università degli Studi dell'Insubria (Italy). She received her master degree in Electronical Engineering in 2002 and her Ph.D. in Computer and Telecommunications Engineering in 2006 from Università degli Studi di Catania (Italy). From September 2004 to March 2006 she has been a research scholar at Politecnico di Milano Since May 2006 she works at Università degli Studi dell'Insubria in the software engineering group.
Her research interests are on wireless sensor networks (WSN), risk assessment methodology and privacy models.
She is a member of the Editorial Board of Computer Network (Elsevier) and IC@ST magazine. She is the general co-chair of S-Cube'09, a steering Committee member of S-Cube' 10 and S-Cube'11, guest editor for the ACM Monet Special Issue, named '' Sensor, system and Software'', TPC member and reviewer for many journals and conferences.