CROSS LAYER DATA ASSESSMENT IN WIRELESS SENSOR NETWORKS

Alberto Coen Porisini and Sabrina Sicari
Dipartimento di Scienze Biologiche, Informatiche e della Comunicazione, Università degli Studi dell'Insubria, via Mazzini 5, Varese 21100, Italy

Keywords: WSN, Privacy, Secure Localization, Data Quality.

Abstract: Wireless sensor networks (WSN) are the target of different kinds of security attacks. The network nodes, which sense, aggregate, encrypt and transmit data, play a key role for assuring data quality. In this paper we present a way in which the network sink can evaluate the nodes reputation in order to determine whether one or more nodes are behaving maliciously. The approach combines different techniques such secure localization and privacy aware transmission in order to assess both nodes reputation and data quality.

1 INTRODUCTION

Wireless Sensor Networks (WSN) technologies support data collection and distributed data processing by means of very small sensing devices (Akyildiz et al., 2007), with limited computation and energy capabilities.

In many applications contexts it is necessary to know the location information of sensor nodes (Akyildiz et al., 2007) and thus, location-aware sensor devices are becoming the *de facto* standard in all domains requiring location-based service. Equipping each sensor with a GPS receiver is not a feasible solution from both an economic and technical perspective since sensors are often deployed in very large numbers and require manual configuration. Thus, position is usually computed by means of nodes cooperation before being transmitted. 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.

Privacy is another crucial issue for many WSN applications such as localization and telemedicine. However, wireless communications and the deployment in uncontrolled environments raise several issues 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, which can be found in literature, are based on access control and strong authentication. However, both techniques are not suitable to WSN because of the limited resources and short battery life. Moreover, approaches based on pre-shared encryption keys are prone to physical attacks since sensor devices and their keys can be easily cloned.

This paper tackles both secure localization and privacy issues following the modeling approach proposed in (Coen-Porisini et al., 2007), (Coen-Porisini et al., 2010) and (Coen-Porisini et al., 2010) 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. Our approach combines together several cheap protection techniques to evaluate the overall data quality. 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. We use privacy compliance and a secure localization protocol to gather information about the data generated by the sensors to assess the overall quality of the data collected by the sink node.

Coen Porisini A. and Sicari S..
CROSS LAYER DATA ASSESSMENT IN WIRELESS SENSOR NETWORKS.
DOI: 10.5220/0003901102030208
In *Proceedings of the 1st International Conference on Sensor Networks* (SENSORNETS-2012), pages 203-208
ISBN: 978-989-8565-01-3

2 FOUNDATIONS

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 (Directive 95/46/EC). 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. A *WSN Privacy Policy* is characterized by three types of classes: *Node, Data, and Action*.

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, 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, 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, while the latter represents the information that is sensed by the nodes of the network. 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 uniquely identified by means of a label *n* and that 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 by means of different messages. 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*.

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

$msg_{n,q}$=<*curr, prv, sub, sensId, errId, errFl, data, idL*>

- *curr*: is the couple <n, q>, which unambiguously identifies the current message among those transmitted by node n.
- *prv*: is a couple $<n_p, q_p>$, where n_p, is the identifier of the node that operated the 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 $<n_s, q_s>$ 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 this field identifies the node that found the error.
- *sensId*: is a couple $<n_{si}, q_{si}>$ 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 $<n_{ei}, q_{ei}>$, 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.

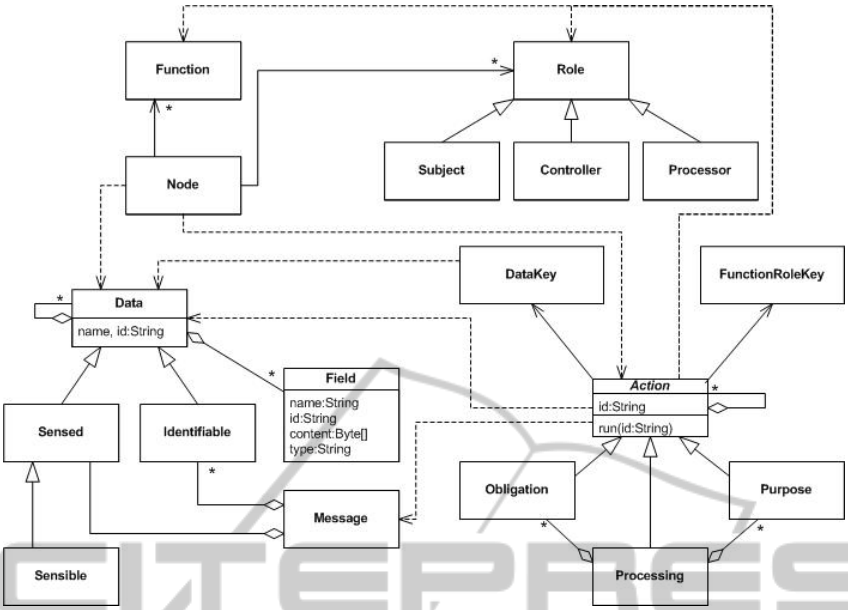

Figure 1: UML Model.

– *errFl*: represents an error code reporting whether an anomaly was identified in the message content.
– *data*: includes the ciphered data sensed by the subject node.
– *idL:* 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 *errFl* equals 1, that is the message reports an error notification.

In order to guarantee the confidentiality of messages content every field but *errFl* 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* is the function-role played by node *n* (The SS key is equivalent to the DataKey of the conceptual model). 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 PROTOCOLS

This section presents the protocols introduced in order to guarantee secure localization and privacy requirements. More specifically the protocols introduced the following:

– *Sensing*, which defines the actions that a node carries out to communicate sensed data;
– *Message Reception and Integrity Verification*, which defines the actions that a node carries out when receiving a message from other nodes;
– *Secure localization*, which defines the action that a node carries out to localize in secure manner;
– *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.

3.1 The Sensing Protocol

Let *n* be a node sensing a data *d* from the environment. Hence 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 previously discussed structure. Notice that, when preparing the message the node acts as a *Transmitter-Processor* (*TP*) and therefore every ciphered field but *data* is encrypted using k(n, TP).

Thus the non-empty fields of $m_{n,q+1}$ are:

curr = sub =<Enc(n,k(n,TP)), Enc(q+1,k(n,TP))>;
data = Enc(d, k(n, SS)); idL = {Enc(n, k(n,TP))};
errFl = 0.

Once prepared part of the message (fields *data* and *sub*) 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 *prv* 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.

curr = <Enc(n,k(n,TP)), Enc(q+1,k(n,TP))>;
prv = $m_{j,h}$*.curr;*
sub = $m_{j,h}$*.sub; sendId =* $m_{j,h}$*.sendId; errFl = 0;*
data = $m_{j,h}$*.data; idL =* $m_{j,h}$*.idL* $\cup${*Enc(n, k(n,TP))*}.

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 node from which it received the message preserved the integrity of the 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.

curr = sub =<Enc(n,k(n,TP)), Enc(q+1,k(n,TP))>;
sendId = retrieved.sub; errId = $m_{j,h}$*.curr. errFl = 1;*
data = Enc(retrieved.data, k(n,NC));
idL = $m_{j,h}$*.idL* $\cup${*Enc(n, k(n,TP))*}.

Notice that *errFl* is set to 1 to indicate that the current message is an error message; field *prv* 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; *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 (Capkun and Hubaux, 2006), which uses three or more anchor nodes (verifiers) to detect misbehaving nodes.

Once computed by the verifiers, 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, if the *δ-test* fails then the estimation is considered to be affected by malicious tampering and thus node *u* is marked as *Malicious*. If the *δ-test* is passed node *u* is marked as *Robust* or *Unknown* depending on whether *u* lies inside the triangle formed by the three verifiers.

3.4 Cross-layer Node Evaluation

The sink evaluates the trustworthiness of the nodes of the network by using both looking at the messages it receives and 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 *errFl* 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 *errFl* 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*, *idL and curr*) by means of 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. 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.

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 RELATED WORKS

Designing secure WSN is a very mature research field (an exhaustive and very comprehensive view of this topic can be found in (Chan and Perrig, 2003)). 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 (Gruteser et al., 2003) and privacy policy based approaches (Snekkenes, 2001).

For instance, (Gruteser et al., 2003) proposes a solution that guarantees the anonymous usage of location based information, focusing on localization services and therefore, constrains the middleware architecture required to support the proposed algorithm.

Other approaches belonging to the former solution are K-Anonymity (Samarati and Sweeney, 1998); 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 (Perrig et al., 2002), 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 (Coen-Porisini et al., 2010), (Gruteser and Grunwald, 2003), (Snekkenes, 2001), (Molnar and Wagner, 2004) 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 (Snekkenes, 2001). Other works (Molnar and Wagner, 2004) address the access control and authentication issues, for instance Duri et al.(Duri et al., 2000) propose a policy based framework for protecting sensor information.

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 (Eschenauer and Gligor, 2002), (Pietro et al., 2003), (Pietro et al., 2009), authentication, access control solutions.

5 CONCLUSIONS

Data quality is a fundamental requirement in any

WSN scenario. 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 behaviour.

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 facing privacy and secure localization issues at the same time. CLP definition is supported by means of a UML conceptual model that defines privacy policies in the context of WSN. The model provides the basic concepts involved when dealing with privacy-related information. At the moment we are carrying out simulations in order to evaluate the CLP performance in real settings.

REFERENCES

I. F. Akyildiz, T. Melodia, and K. Chowdhury, "A survey on wireless multimedia sensor networks," *Elsevier Computer* Networks *Journal*, March 2007.

A. Coen-Porisini, P. Colombo, S. Sicari, and A. Trombetta, "A conceptual model for privacy policies," in *Proc. of SEA 2007*, Cambridge (MS), USA.

A. Coen-Porisini, P. Colombo, and S. Sicari, "Dealing with anonymity in wireless sensor networks," in In *Proceedings of 25th annual ACM symposium on Applied Computing (ACM SAC)*, Sierre, Switzerland, 2010.

A. Coen-Porisini, P. Colombo, and S. Sicari, "Privacy aware systems: from models to patterns" in Software Engineering for Secure Systems: Industrial and Research Perspectives, IGI Global, editor Dr. H. Mouratidis, 2010

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

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

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

Q. Ni, A. Trombetta, E. Bertino, and J. Lobo, "Privacy-aware role based access control," in *Proceedings of the 12th ACM symposium on Access control models and technologies*, 2007.

H. Zhanga, A. Arorab, Y. Choic, and M. Goudac, "Reliable bursty convergecast in wireless sensor networks," *Elsevier Computer Communications*, vol. 30, no. 13, pp. 2560–2576, 2007.

OMNeT++ Discrete Event Simulation System. http://www.omnetpp.org/doc/manual/usman.html, 2005.

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

H. Chan and A. Perrig, "Security and privacy in sensor networks," *IEEE Computer Magazine*, pp. 103–105, March 2003.

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

P. Samarati and 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.

M. Gruteser and 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.

A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, "Spins: security protocols for sensor networks," *Wireless Networking*, vol. 8, no. 5, pp. 521–534, 2002.

E. Snekkenes, "Concepts for personal location privacy policies," in In *Proceedings of 3rd ACM Conf. on Electronic Commerce*, 2001.

D. Molnar and D. Wagner, "Privacy and security in library rfid: Issues, practices, and architectures," in In *Proceedings of ACM CCS*, 2004.

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

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

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

S. Capkun and J. Hubaux, "Secure positioning in wireless networks," in *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 221–232, Feb. 2006.

S. Brands and D. Chaum, "Distance-bounding protocols," in *Proceedings of Workshop on the Theory and Application of Cryptographic Techniques*, 1994.