Assessing Data Quality by a Cross-Layer Approach

Mattia Monga
Dip. di Informatica e Comunicazione
Università degli Studi di Milano
Via Comelico 39, I–20135 Milano, Italy
Email: mattia.monga@unimi.it

Sabrina Sicari
Dip. di Informatica e Comunicazione
Università degli Studi dell'Insubria
Via J.H.Dunant 3, I–21100 Varese, Italy
Email: sabrina.sicari@uninsubria.it

Abstract

Wireless sensor networks had a big diffusion in the last few decades and they are used in many application domains. Services built on them require to handle a big amount of data, and a fundamental requirement is their quality, highly affected by the security of the whole system. Some encryption techniques can be adopted, but it is also necessary to verify the reliability of nodes that sense, aggregate, transmit data and share keys. This paper proposes a methodology for assessing data quality. Built on a cross layer approach, our methodology exploits localization information.

Keywords

Wireless sensor network, security, localization, distributed aggregation

1. Introduction

Wireless sensor networks (WSNs) are becoming popular in many application domains: indoor/outdoor surveillance systems, traffic monitoring and control systems for urban and sub-urban areas, systems supporting tele-medicine, attendance to disable or elderly people, environment monitoring, localization and recognition of services and users, monitoring and control of manufacturing processes in industry, etc. [1]. The amount of data that have to be transmitted and manipulated to provide such complex services is pretty high, but sensors are tiny devices with limited computation and energy capabilities and the transmission of data is a very expensive operation from a power consumption perspective. Thus, one of the design goals of WSN systems should be the reduction of the need for transmission. A step in this direction are publish-subscribe approaches and content-based routing [2], [3]. In several situations, however, the interest is just in some *aggregated* measure, such as the average temperature of a region, the average humidity, etc. In this cases a proper aggregation algorithm (for examples see [4], [5], [6]) may reduce significantly the number of bytes exchanged across the WSN.

From a security point of view, the wireless communications and the deployment in uncontrolled environments rise several issues: in fact, the confidentiality, the integrity, and the availability of data might be put at risk by malicious tampering of sensors and/or traffic. Many of the solutions proposed in the literature are based on access control and strong authentication, which are problematic to implement with limited resources and short battery life. Moreover, the approaches based on pre-shared encryption keys are prone to physical attacks: in fact, is pretty easy to clone a sensor device and its key. Therefore, in this paper we start from the assumption that the perfect protection of data is infeasible or too expensive to be effective and we propose to use an approach based on the assessment of data quality, defined as how likely is that the information to be elaborated by the application is reliable and trustworthy. Therefore, we suggest to combine two — in general unreliable, but relatively easy to achieve — protection techniques: while none of them is enough to guarantee the reliability of all the pieces of information, consistency across them can

9781-4244-3941-6/09/$25.00 © 2009 IEEE

be exploited to get a measure of the overall quality. Applications can then leverage on this assessment to provide the level of service feasible in the current context. The aim of our work is to define a methodology that defines the fundamental steps for assessing data quality. We use secure data aggregation and a secure localization protocol to gather information about the data generated by the sensors and we compute a cross-layer assessment of the overall quality of the data collected by the sink node. The paper is organized as follows: Section 2 provides a short state of art about data aggregation in hostile environments; Section 3 shortly describes the reference scenario in order to clarify the application domain; Section 4 introduces the main steps of the methodology and some implementation details. Section 5 draws some conclusions and provides hints for future works.

2. Related Work on Secure Aggregation

Data confidentiality and integrity become vital when sensor nodes are deployed in a hostile environment. Secure aggregation of data in a adversarial setting is the focus of several works. They can be classified according to the approach used for encrypting data, that can be done *hop-by-hop* or *end-to-end.* In the former case, the data are encrypted by sensing nodes and decrypted by aggregators. The aggregator nodes, then, decrypt the data coming from the sensing nodes, aggregate them, and encrypt them again, until eventually the sink node gets the final encrypted aggregation result (and decrypts it). In the end-to-end approach the intermediate aggregators manipulate only encrypted data and they have no keys to decrypt them.

Some hop-by-hop works (for example, [5], [7], [8], [6]) assume that data security is guaranteed by means of some key distribution schemes. Such solutions are vulnerable to the tampering or cloning of the intermediate aggregator nodes. End-to-end encrypted techniques ([9], [4]) partially overcome the weakness of hop-by-hop techniques, but they also need a key scheme. Some approaches suggest to share a key among all sensing nodes and the sink, however, the aggregators have no keys to manage because they handle data without making any decryption. Nevertheless, sensing nodes' keys are still critical and if they are shared among too many nodes, a whole network region might be compromised by attacking a single sensing node. An alternative is represented by the adoption of public key encryption [10], but in this case the drawback is represented by a high computational power consumption.

Most of the proposed solutions are based on the adoption of encryption techniques, ad-hoc key distribution schemes [11], [12], [13], authentication, access control solutions. Our methodology proposes, instead, to combine some cheap protection techniques even if none of them is totally effective. We leverage on cross-layer consistency to asses the overall quality of the collected information. Thus, by means of the verification of localization information it is possible to adopt an end-to-end secure data aggregation with a pre-shared key, without wasting power with public key encryption techniques.

3. Reference Scenario

We consider a dense network composed of nodes n_i, where $n_i \in N, 0 < i \leq |N|$ and a base station b in which all the collected data sink. We consider three subsets of N:

- S, composed by nodes $s_i, 0 < i \leq |S|$ which perform sensing functions;
- A, composed by nodes $a_i, 0 < i \leq |A|$ which aggregate data;
- V, composed by nodes $v_i, 0 < i \leq |V|$ which work as verifiers in the secure localization protocol.

$N = S \cup A \cup V$, however while V may overlap both S and A (in principle every node whose position can be taken for granted might be used as a verifier), the effectiveness of secure aggregation requires (see Section 4.1) that S and A are disjoint, *i.e.,* $S \cap A = \emptyset$.

Each s_i node senses a given type of data (*e.g.,* temperature, pressure, brightness, position, and so on), while a_i nodes combine the sensing data received from sensing nodes in their communication range. Each (sensing, aggregator, and verifier) node directly communicates with its closer neighbours (at one hop distance).

All the sensors s_i whose data are to be collected by the node a_j share a symmetric encryption key κ_{a_j} with b. As we will discuss in Section 4.1, there is no need for storing any key on aggregator nodes.

4. Assessing the quality of collected data

We combine secure data aggregation and a secure localization protocol to gather information about the data generated by the sensors. Thus, the sink node b is able to compute a cross-layer assessment of the overall quality of the collected data. The proposed strategy is composed by four steps:

1) *Secure data aggregation.* s_i nodes send their sensed data to the nearest a_j. The data are encrypted with a key κ_{a_j} not known to a_j. The a_j nodes aggregate all the received data *without decrypting them* (see Section 4.1) and send them to b. At the end of the process, b has the data in their aggregated form: for example the average temperature of a region. Encryption guarantees the integrity and confidentiality of the transmitted data, while the key is not known to attackers. However, a clone of a sensor could have sent a forged datum that nonetheless was aggregated with the others.
2) *Node localization.* s_i nodes provide to v_j the pieces of information needed to localize them. A beacon signal is normally enough: the algorithm we chose (see Section 4.2) is based on round-trip time measurement and the v_j nodes end up with a *distance bound* to s_i; distance bounds are then sent to b. A malicious node might delay the beacon signal, but not speed it up: as a consequence the distance bounds could be larger than the actual ones, but not shorter. It is worth noting that the localization data cannot be transmitted as described in the previous step, since no aggregation is performed.
3) *Assessment of localization.* b, thanks to the distance bounds received from the v_j nodes (at least three for each s_i are needed), can now compute the position[1] (x_i, y_i) of each sensing node s_i; moreover the verifiable multilateration (see Section 4.2) provides a measure W_i for the trustworthiness of the result: each position can be marked as `Robust`, `Malicious`, or `Unknown`. At the end, b has a table $T = < (x_i, y_i), W_i >$ that maps each position to its reliability measure.
4) *Cross-layer assessment of data quality.* b can use T to assess the quality of data aggregated on a given region Ξ; in fact, for each position *within* Ξ, a measure of its reliability is known: thus, according to the constraint of the application domain, b may decide to discard the aggregated data if it is not considered reliable enough.

The next sections analyze in depth the previous steps.

4.1. Secure Data Aggregation

Limitations on power require a minimization of the amount of transmitted data from nodes to the sink. Aggregation protocols may help in reducing the overall traffic among nodes. At the same time, since nodes are the attack goals of malicious users which try to violate the confidentiality and the integrity of data, proper countermeasures are needed to perform a secure data aggregation. Encryption can be used to secure node communication both hop-by-hop and end-to-end (see Section 2). We chose an end-to-end secure aggregation solution in which an attack to any aggregator node is not able to compromise the whole system. We adopted the algorithm described in [4] because it is based on a simple and secure additively homomorphic stream cipher that allows efficient aggregation of encrypted data. The cipher algorithm uses modular additions and is therefore very well suited for CPU-constrained devices. Aggregation based on this cipher can be used to efficiently compute statistical values such as mean, variance, and standard deviation of sensed data, while achieving a significant bandwidth gain. A *homomorphic encryption scheme* enables arithmetic operations to be performed on encrypted data. One example is a multiplicatively homomorphic scheme, whereby the multiplication of two ciphertexts followed by a decryption operation yields the same result as, say, the multiplication of the two corresponding plain-text values. 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. A stream cipher is typically obtained by combining plain-text bits with a pseudo-random cipher bit stream (key-stream) by an exclusive-or ($\oplus$) operation. Since $(a \oplus k) \oplus k = a$, decryption is also easy to obtain. The main idea of the homomorphic encryption described in [4] is to replace the xor with modular addition $(a + b) \, mod \, M$. The main steps are the following:

[1] In this paper we consider nodes as points in the 2D space

- *Encryption*
  - Each datum is represented by an integer $0 \leq d \leq M-1$, where M is a large integer;
  - Let k be a randomly generated keystream, where $0 \leq d \leq M-1$
  - The cipher-text c is given by $c = Enc(d,k,M) = (d+k)\, mod\, M$
- *Decryption:* $d = Dec(c,k,M) = (c-k)\, mod\, M$
- *Addition of cipher-texts*
  - Let $c_1 = Enc(d_1,k_1,M)$ and $c_2 = Enc(d_2,k_2,M)$
  - then, if M is sufficiently large, $Dec(c_1 + c_2, k, M) = d_1 + d_2$, where $k = (k_1 + k_2)\, mod\, M$

The scheme is made practical by generating encryption keys in each session with a pseudo-random function applied to a unique node id (see [4] for the details): from a logical point of view, the sensor nodes and the sink share the encryption key of the aggregated datum. However, by attacking or cloning a single sensor node, an enemy cannot compromise the whole system, but at worst only aggregate a fake datum. Aggregator nodes, instead, are immune from attacks, since they handle only encrypted data.

4.2. Secure Localization

The node positions can be evaluated by using a multilateration technique, which determines the node coordinates by exploiting a set of landmark nodes, called the *anchor* nodes, whose positions are known. The position of the unknown node u is computed by using an estimation of the distances between the anchor nodes and the node itself. The distance is not measured directly; instead, it can be computed by knowing the speed of the signal in the medium used in the transmission, and by measuring the time needed to get an answer to a beacon message sent to u. If the computation is carried on without any precaution, u might fool the anchors by delaying the beacon message. However, since a malicious node can delay the answer beacon, but not speed it up, under some conditions it is possible to spot malicious behaviors. *Verifiable Multilateration* (VM) [14] uses three or more anchor nodes to detect misbehaving nodes. In VM the anchor nodes work as *verifiers* of the localization data and they send to the sink b the information needed to evaluate the consistency of the coordinates computed for u. The basic idea of VM is shown in Figure 1: each verifier v_i computes its *distance bound* [15] to u; any point $u' \neq u$ inside the triangle formed by v_1, v_2, v_3 has necessarily at least one of the distance to the v_i enlarged. This enlargement, however, cannot be masked by u by sending a faster message to the corresponding verifier. Therefore, if the verifiers are trusted and they can communicate securely with b, the following algorithm can be used to check the localization data:

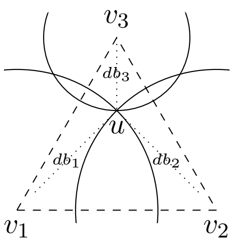


Figure 1. Verifiable multilateration

1) Each verifier v_i send a beacon message to u and records the time τ_i needed to get an answer;
2) Each verifier v_i (whose coordinates $< x_i, y_i >$ are known) send to b a message with its τ_i;
3) From τ_i, b derives the corresponding distance bound db_i and it estimates u's coordinates by minimizing the mean square error $\epsilon = \sum_i (db_i - \sqrt{(x_u - x_i)^2 + (y_u - y_i)^2})^2$, where $< x_u, y_u >$ are the coordinates to be estimated[2];
4) b can now check if $< x_u, y_u >$ are feasible in the given setting by two incremental tests:
   a) *δ-test:* For all verifiers v_i, compute the distance between the estimated u and v_i: if it differs from the measured distance bound by more than the expected distance measurement error, the estimation is affected by malicious tampering;
   b) *Point in the triangle test:* Distance bounds are reliable only if the estimated u is within at least one verification triangle formed by a triplet of verifiers, otherwise the estimation is considered unverified.

[2] Such minimization is indeed far from trivial: we used an approximation based on simulated annealing.

If both the δ and the point-in-the-triangle tests are positive, the distance bounds are consistent with the estimated node position, which moreover falls in at least one verification triangle. Thus, the sink can consider the estimated position of the node as `Robust`; otherwise, the information at hands is not sufficient to support the reliability of the data. An estimation that does not pass the δ test is considered `Malicious`. A sensible value of δ depends on the expected error in time measurement and the number of available verifiers. The simulation reported below should clarify the considerations involved in the choice of δ. If the δ test is passed, but the point-in-the-triangle one fails, the sink marks the estimation as `Unknown`, meaning there is no sufficient information for evaluating the trustworthiness of node position. Thus, the localization phase ends up, for each unlocalized node u_i, with an estimation of the position of u_i and a quality $W_i \in Robust, Unknown, Malicious$.

4.2.1. Simulation. We used OMNET++ (ver. 3.3p1, [16], [17]) to set up a simulation of the secure localization algorithm. A claimant node u to be localized resides at the center of a $100\text{m}\times 100\text{m}$ field, *i.e.,* at point $< 50, 50 >$. Since the best approach to lay out three verifiers is on the vertexes of an equilateral triangle [14], we fixed their coordinates to be the points $< 1, 1 >, < 99, 1 >, < 50, 85 >$. If u is *faithful,* it answers to verifiers' beacons without any delay. Otherwise, if u is *malicious* it adds a variable delay to the answers, in order to dissimulate a fake position u': *i.e.,* for each v_i, if the distance $\overline{v_iu'}$ is greater than $\overline{v_iu}$ a proper delay is added by u to the answer beacon to v_i. We assumed that signals travel at the speed of light and that time can be measured with an error whose standard deviation is 2ns. As described above, the timing information collected by verifiers v_i can be used by the base station to classify the claimant as `Malicious`, `Unknown`, or `Robust`. Figure 2(a) shows the effect of the choice of the δ_{max} in the δ-test on 10000 runs with 3 verifiers: the only sensible value is 35, since lower levels have an overwhelming rate of false positives (*i.e.,* faithful nodes classified as `Malicious`), and a higher δ gives too much false negatives (*i.e.,* malicious nodes classified as `Robust`) and unknowns. About 50% of malicious claimants and 90% of faithful ones were classified as `Unknown`: the error in taking the estimated position instead of the real one is pretty high, as one can see from Figure 3 that plots the Empirical Cumulative Distribution Function of deception. The situation is clearly improved when a fourth verifier is added (see Figure 2(b)): the setting is now with a verifier at each corner of the field and all the values less than 2.5 give acceptable results; there are no `Unknowns`. It is worth noting that the range of δ considered is different, since by increasing the number of verifiers, the maximum acceptable error δ_{max} should decrease. There are still some false negatives, but the deception induced by a malicious node taken as `Robust` is always less than 1m with $\delta \leq 1$. Figure 4 plots the density distribution of the deception — *i.e.,* the distance between the real position and the estimated one — at different values of δ. Adding a fifth verifier randomly deployed significantly decreases the rate of false negatives, as shown in Figure 2(c).

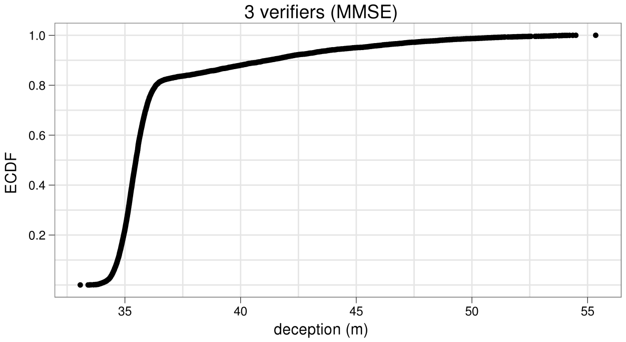


Figure 3. Empirical Cumulative Distribution Function (ECDF) of deception for `Unknown` nodes

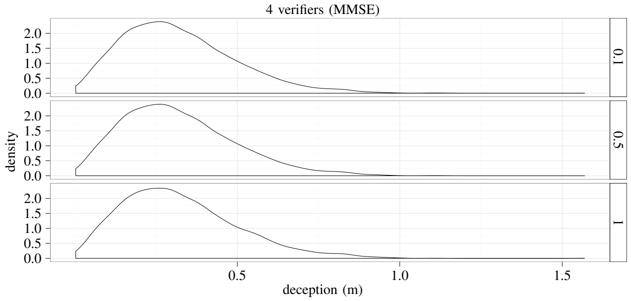
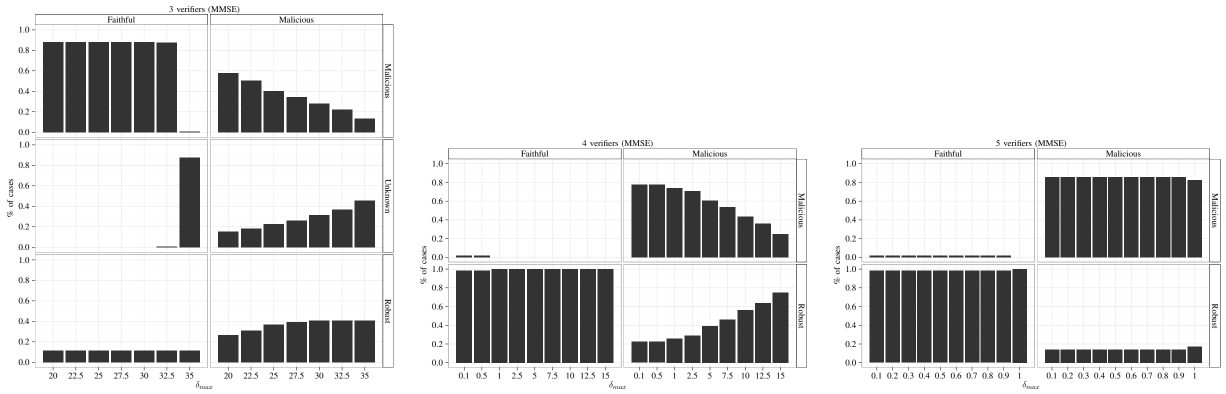


Figure 4. Deception when a malicious node is classified as `Robust`



(a) 3 verifiers on a centered equilateral triangle

(b) 4 verifiers on field corners

(c) 5 verifiers: 4 on field corners, 1 randomly deployed

Figure 2. Classification by secure localization

4.3. Cross-layer Assessment of Data Quality

The sink receives a message encrypted by the scheme sketched in Section 4.1. Once decrypted, the sink has the aggregated datum, together with the guarantee of its integrity and confidentiality. It could, however, embed fake data coming from malicious nodes which had mocked faithful nodes. The sink has also data about the localization of nodes, together with a marking of their trustworthiness quality. Since each aggregated datum d *refers to a given region* Ξ_i, in order to assess the quality of d, the sink might use the localization information about any node $n_i \in \Xi_i$ and each node $n_j \in \Xi_j, j \neq i$ whose quality is `Malicious` or `Unknown` and the region Ξ_j confines with the region Ξ_i. In other words, the sink can use cross-layer information to guess how many possibly fake data were deceptively aggregated in the final result. It is still possible that an attacker decided to not lie on its position, but only in the datum sent to the aggregator. In that case, however, the position *is* faithful and other consistency properties could be exploited: for example, since temperature is a continuous quantity, a datum of 40°C could be found as anomalous if its close to other measures that are about 20°C. Unfortunately, this kind of considerations are application specific and no general rule can be given. In specific cases, a fake datum associated to a truthful localization can always snake in; however, at least the true position might be used to spot the malicious sensor. All in all, the cross-layer analysis enables a more careful assessment of the overall quality of the received data, and data possibly affected by too many attackers can be discarded, thus avoiding malicious poisoning.

Use case

A simple numerical example should be sufficient to illustrate the application of the proposed approach.

A base station b has received the average temperature (25°C) of a rectangular region defined by the four points $< 0,0 > - < 10,0 > - < 10,10 > - < 10,0 >$. b has built the following table listing all the information received about sensor nodes localization:

x_i	y_i	W_i
2	3	`Robust`
4	5	`Robust`
5	2	`Malicious`
11	3	`Malicious`
12	4	`Robust`

Therefore, b may now asses the quality of the aggregated data received: thus, the average 25°C contains the data coming from the two sensors positioned in $< 2,3 >$ and $< 4,5 >$, and possibly from the two sensors positioned in $< 5,2 >$ (maliciously asserting to be inside the region, but probably outside) and $< 11,3 >$ (maliciously asserting to be outside the region, but possibly inside). Since the datum could be affected by two malicious nodes, b decides to discard it.

5. Conclusions

Data quality is a fundamental requirement in any wireless sensor network scenario. Although it is very difficult to provide data trustworthiness due to the distributed nature and the limited resource in terms of power of WSN, the proposed methodology allows to analyze data trustworthiness by exploiting consistency on cross-layer information, *i.e.,* node localization and data aggregation. The proposed solution improves the knowledge about the security behavior of nodes that handle data. More specifically, the trustworthiness about the node position information is used as a metrics for evaluating data trustworthiness. In fact node position, being target of different kind of attacks — i.e., node malicious displacement, or distance enlargement — is a good alarm for revealing malicious behavior. Our methodology is flexible, in fact it results largely independent from the adopted routing protocols, the verification localization algorithm, and the secure data aggregation strategy. Our approach can also be applied on wireless multimedia sensor networks, due to its independence from the kind of sensing data (multimedia or monomedia data). At the moment possible extensions for modelling privacy policies and the related enforcement mechanisms are under investigation. Moreover the application of game theory for modelling malicious behavior and reason about rational choices, represents a future goal.

Acknowledgment

This research has been partially funded by the European Commission, Programme IDEAS-ERC, Project 227977-SMScom.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on wireless sensor network,” *IEEE Wireless Communications*, vol. 40, no. 8, pp. 102 – 114, August 2002.

[2] Patrick, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, “The many faces of publish/subscribe,” *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114–131, June 2003. [Online]. Available: http://dx.doi.org/10.1145/857076.857078

[3] G. Cugola, A. Margara, and M. Migliavacca, “Context-aware publish-subscribe: Model, implementation, and evaluation,” in *Proc. of the IEEE Symposium on Computers and Communications (ISCC'09)*, Sousse, Tunisia, 2009.

[4] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, “Efficient and provably secure aggregation of encrypted data in wireless sensor networks,” *ACM Trans. Sen. Netw.*, vol. 5, no. 3, pp. 1–36, 2009.

[5] L. Hu and D. Evans, “Secure data aggregation in wireless sensor networks,” in *Workshop on Security and Assurance in Ad hoc Networks*, 2003.

[6] M. Bagaa, N. Lasla, A. Ouadjaout, and Y. Challal, “Sedan: Secure and efficient protocol for data aggregation in wireless sensor networks,” in *Proc. of 32nd IEEE Conference on Local Computer Networks*, 2007.

[7] A. Mahimkar and T. Rappaport, “Securedav: A secure data aggregation and verification protocol for sensor networks,” in *Proc. of IEEE Globecom*, Dallas, Tx, 2004.

[8] B. Przydatek, D. Song, and A. Perrig, “Sia: Secure information aggregation in sensor networks,” in *Proc. of ACM SenSys*, 2003.

[9] J.Girao, D.westhoff, and M. Schneider, “Cda: concealed data aggregation for reverse multicast traffic in wireless sensor networks,” in *Proc. of IEEE ICC*, Korea, 2005.

[10] E.Mykletun, J.Girao, and D. Westhoff, “Public key based cryptoschemes for data concealment in wireless sensor networks,” in *Proc. of IEEE ICC*, Turkey, 2006.

[11] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks.” in *Proc. of 9th ACM Conference on Computer and Communications Security*, 2002.

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

[13] R. D. Pietro, C. Soriente, A. Spognardi, and G. Tsudik, “Collaborative authentication in unattended wsns,” in *Proc. of 2nd ACM Conference on Wireless Network Security (WiSec*, Zurich, Switzerland, 2009.

[14] S. Čapkun and J. Hubaux, “Secure positioning in wireless networks,” *IEEE Journal On Selected Areas In Communications*, vol. 24, no. 2, pp. 221–232, Feb. 2006.

[15] S. Brands and D. Chaum, “Distance-bounding protocols,” in *Proc. of Workshop on the Theory and Application of Cryptographic Techniques*, 1994.

[16] OMNeT++ Community, http://www.omnetpp.org/.

[17] G. Pongor, “OMNeT: objective modular network testbed,” in *MASCOTS’93 Proc. of the International Workshop on Modeling, Analysis, and Simulation On Computer and Telecommunication Systems*. San Diego, CA, USA: The Society for Computer Simulation, International, 1993, pp. 323–326.