Computer Communications 34 (2011) 515–523

Contents lists available at ScienceDirect

Computer Communications

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

Location privacy and resilience in wireless sensor networks querying

Roberto Di Pietro [a,*], Alexandre Viejo [b]

[a] Università di Roma Tre, Dipartimento di Matematica, L.go S. Leonardo Murialdo n.1, 00146 Roma, Italy
[b] Departament d'Enginyeria Informàtica i Matemàtiques, UNESCO Chair in Data Privacy, Universitat Rovira i Virgili, Av. Països Catalans 26, E-43007 Tarragona, Spain

ARTICLE INFO

Article history:
Available online 8 June 2010

Keywords:
Location privacy
Probabilistic algorithm
Resiliency
Security
Wireless sensor networks

ABSTRACT

Due to the wireless nature of communication in sensor networks, the communication patterns between sensors could be leaked regardless of the adoption of encryption mechanisms—those would just protect the message content. However, communication patterns could provide valuable information to an adversary. For instance, this is the case when sensors reply to a query broadcast by a Base Station (BS); an adversary eavesdropping the communication traffic could realize which sensors are the ones that possibly match the query (that is, the ones that replied). This issue is complicated by the severe resource constrained environment WSNs are subject to, that call for efficient and scalable solutions.

In this paper, we have addressed the problem of preserving the location privacy of the sensors of a wireless sensor network when they send a reply to a query broadcast by the BS. In particular, we deal with one of the worst scenarios for privacy: When sensors are queried by a BS to provide the MAX of their stored readings. We provide a probabilistic and scalable protocol to compute the MAX that enjoys the following features: (i) it guarantees the location privacy of the sensors replying to the query; (ii) it is resilient to an active adversary willing to alter the readings sent by the sensors; and, (iii) it allows to trade-off the accuracy of the result with (a small) overhead increase. Finally, extensive simulations support our analysis, showing the quality of our proposal.

1. Introduction

A wireless sensor network (WSN) is a network composed of resource constrained devices with sensing capabilities and enabled with wireless communications technologies. These small devices are called *sensors*. WSNs have a broad range of applications including both military and civilian usage. These applications are mainly related to surveillance and habitat monitoring. In this kind of scenarios, hundreds of sensors are usually randomly scattered around the area that must be monitored. After deployment, sensors form a WSN in an ad hoc manner and finally, the whole network starts its activity [1].

A common monitoring application in WSN involves a large amount of sensors sending sensed data towards a Base Station (BS) that processes all the received information. This situation corresponds to the *many-to-one* paradigm. This paradigm follows a tree communication topology where the leaves of the tree act as senders and the root of the tree is the only receiver. In this scenario, the root is very likely to be swamped when too many leaves transmit simultaneously. This problem is known as implosion [35].

In addition to that, due to the limitation in resources, WSNs require protocols that are efficient in the usage of both computations and communications, to save energy. In [39], a general framework for scalable many-to-one communication is presented. In that framework, intermediate sensors collect messages from their children, aggregate them and send a single aggregated message up to their parent. In this way, the root receives a single aggregated message. This process is called *data aggregation*. Data aggregation solves the implosion problem and allows the sensors of the WSN to preserve their resources.

Privacy issues in WSN could be ascribed to two main fields.

As for the first one, preserving privacy is important in applications where the behavior of individuals or businesses can be deduced from sensed data in a monitored environment. It is often easy to deduce sensitive information based on seemingly innocuous sensed data. For instance, the behavior of a household can be tracked or profiled by monitoring their electrical usage [21]. Since sensing is passive, it can be done without the knowledge of the observed party or environment—the observed party has very limited ability to control the extent of information that is disclosed by the sensors. Hence, maintaining the *privacy of sensed data* can be important to parties in the sensed environment. Similarly, sensor-data privacy can be important to a service provider that provides service based on the status of the network-sensed data. The service provider may want to guarantee the privacy of the observed party

* Corresponding author. Tel.: +39 06 5733 8246.
E-mail addresses: dipietro@mat.uniroma3.it (R. Di Pietro), alexandre.viejo@urv.cat (A. Viejo).

0140-3664/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2010.05.014

as it can face liability from regulators and customers. Hence both service providers and their customers must rely on the network owner/operator to provide sensor-data privacy. Unless privacy challenges are addressed by the sensor network owner/operator, wide-spread adoption of WSNs in many applications could be delayed.

As for the second field, it is strictly related to the WSN operating scenario, where the communication patterns of sensors can expose sensitive contextual information. This is the concept of *location privacy*. For example, when a sensor delivers its data to the BS, it may disclose the location of some critical events in the field, revealing valuable intelligence. If the WSN is operating in a hostile environments, location privacy is of paramount importance. Indeed, if this requirements fails, location-based information can completely jeopardize network applications. A straightforward example comes from the military: Disclosure of the locations of soldiers due to nearby sensors communicating with the BS results in a compromise of both their life and the mission. However, providing location privacy in a WSN is a challenging task. On the one hand, an adversary (ADV) can easily intercept the network traffic due to the use of the broadcast nature of transmission. She can then perform traffic analysis and identify the source sensor that initiates the communication with the Base Station [20]. This can reveal the locations of the sensors being monitored or queried by the BS, hence exposing valuable information. On the other hand, the scarce resources sensors are equipped with, make unfeasible the adoption of traditional techniques—used in ground based radio stations that are not subject to the constraints sensors are subject to—to hide the communication from a sensor to the BS [14].

From the above discussion, it follows that in the context of location privacy, it is not the content of the communication that needs to be kept private, but the sensor (or set of sensors) that provided data to the BS—either because an event of interest was detected, or because queried, by the BS. The proposal in this paper addresses this latter issue, presenting a protocol that, other than providing sensor privacy, is also resilient to an active ADV aiming at disrupting the distributed computation. Further, the proposed mechanism introduces limited overhead only.

Usually, applications of sensor networks focus on obtaining aggregate statistics such as SUM, AVERAGE, or MAX/MIN of data readings. These values can be provided proactively by the sensors to the sink—usually on a time based period—or when so requested by the BS via specific query. In some context, as shown above, preserving *location privacy* becomes an important concern. Moreover, in addition to location privacy, *resilience* is another important property to be enforced when contributory computations are processed. In this context, resilience assures that the overall computation is not disrupted by a subset of sensors compromised by ADV. A weaker form of resilience is that, if ADV succeeds in disrupting such a computation (via compromised sensors), the IDs of the compromised sensors can be later exposed if some follow-up activity is undertaken by the BS.

Finally, note that when designing querying protocols that offer location privacy and resilience in WSNs, the following issues must be taken into account: (i) sensors have no powerful computational capabilities—hence resource-demanding cryptography, such as homomorphic public-key cryptosystems, is usually not viable; and (ii) sensors are prone to be captured in a hostile environment—hence resilience, as above introduced, should be provided.

1.1. Contributions

In this paper we provide a probabilistic protocol that allows the BS to query the WSN in order to obtain the MAX value sensed by its constituency sensors. The main feature of this proposal is that it preserves the location privacy of the participating sensors. That is, even assuming an ADV that can monitor the whole network, it cannot infer any information about the sensors that have the value queried by the BS. Moreover, we show that the proposed protocol is also resilient to an ADV that actively tries to disrupt the computation.

The proposed solution enjoys several features: It is lightweight and scalable; communication overhead is evenly distributed among sensors, hence increasing the life-time of the WSN; and, the accuracy of the provided result can be adjusted with just a slight increase in overhead. Finally, extensive simulations support our proposal. Note that it is straightforward to extend the proposed protocol to compute the MIN, as well as to support range-query.

1.2. Roadmap

In next section we introduce the related work while in Section 3 we introduce the background and the assumptions used throughout the paper. In particular, we introduce the adversarial model, the network setup, and the query model. In Section 4 we detail our new protocol while in Section 5 we analyze its security. In Section 6 we provide simulations results of our scheme and discuss these results—proving that the designed protocol is efficient, while providing location privacy and resilience. Finally, Section 7 reports some concluding remarks and highlights future works in the field.

2. Related work

Providing privacy to a WSN strictly depends on the application running on top of the network. Approaches, such as [22,18], are focused on solving the problem of a privacy-preserving data aggregation [40,26]. They highlight an important concern: The need of preserving the location of the sender. Such a problem is initially coped by means of schemes targeting the routing algorithm. Several works try to hide the source of a message [25,32]. The message is initially forwarded to the sink by means of a random walk; then, before the actual deliver, a large amount of fake traffic among a subset of sensors is produced. Under the assumption of an adversary close to the sink (thus having a local view of the network), the source location is deceived. In [8], Conner et al. explicitly protect the location of the sink (a notable extension is discussed in [24]). The adopted algorithm is in charge of luring the traffic to a random sensor; only after the traffic has been aggregated, a final communication to the sink takes place. However, the adversary taken into account has a very local view of the network topology. The first work on location privacy in sensor networks under a global eavesdropper is provided in [28]. This assumption is leveraged and further developed with the work of Yang et al. [41]: To face a global adversary, dummy traffic is no longer an option. In their paper they introduce the concept of carefully chosen dummy traffic to disguise the event source location. They also formalize the concept of event unobservability with respect to wireless communication. To mitigate the produced communication overhead, they propose and discuss the creation of special entities (i.e. proxies) in charge of aggregating the communication addressed to the sink. By a careful placement of the proxies, they argue that the dummy traffic is therefore reduced to a minimum. Likewise, without the proxies addition, in [31] the authors propose a statistically sound scheme where the real messages are always mixed with fake messages. It is not clear though if an attacker is able to identify a real message (even if encrypted) by tracing it back to the source.

As for privacy in contributory computations, the authors in [42] propose a set of privacy-preserving data aggregation schemes to support queries targeted at sensor data. Even though the cited paper focuses on querying the median value of all sensor readings, the authors claim that their proposal can be extended to handle

minimum/maximum queries as well. In their proposal, each sensor data item is an integer ranging from 0 to some upper bound denoted as $P-1$. The idea beneath this set of schemes is to divide $(0,\ldots,P-1)$ in a set of ranges. Upon receiving a query from the Base Station, each sensor node answers in which range is located its reading. This solution reduces the communication overhead when querying certain aggregation functions (*e.g.* the median). Nevertheless, if accurate minimum/maximum queries are performed this solution must be set to work with P different ranges. However, the use of such accurate ranges increases the communication overhead of the system. In addition to the communication cost, the set of protocols presented in [42] do not provide any security against compromised sensor nodes trying to prevent the BS from obtaining the correct value. More specifically, any node of the WSN can easily modify the contributions linked to each range, while the system does not have any mechanism to avoid that.

Other related work aims to achieve anonymity in WSNs by obfuscating sensors' identifiers. The work in [37] is the first attempt in this direction, with the assumption that sensors are anonymous, i.e., a sensor has no initial unique identifier. Sensors are organized in clusters and are aware of their own location. The attacker wants to identify and eliminate the minimum number of sensors to inflict maximum loss of data. Two anonymous schemes for clustered WSNs are proposed in [29] to provide anonymity, assuming secret keys shared between sensors and the Base Station are not compromised. Moreover, sensors in a cluster are considered indistinguishable. Anonymous routing is performed by using pseudonyms to keep sensor identity private in route requests and replies. More recently, Carbunar et al. [4] presented a scheme that attempts providing query privacy in WSNs. The scheme uses two servers. If these two servers collude, privacy is completely compromised. Furthermore, each sensor needs to store a key for each client, thus resulting in a linear increase in storage overhead. Finally, a scheme for anonymous data collection in WSNs was presented in [23]. The security of this scheme is based on data perturbation, rather than on provably secure cryptographic techniques. Finally, a survey of open issues on event unobservability is provided in [30].

An interesting proposal that appeared in [12] introduces the possibility to select the appropriate privacy-preserving policy when querying a WSN. There different techniques supporting the described goal are detailed. A recent privacy issue in WSNs relates to the privacy of the ID of sensors that signal an event is addressed in [11], highlighting a technique focusing on preserving the anonymity of the sensor ID that provides the data of interest, rather than on the privacy of the data itself. However, the protocol provided relies on onion-routing like solution, hence introducing a non-negligible burden on both computational and communication. Further, the level of privacy achieved still needs to be fully validated.

3. System model

3.1. Query model

In this paper we follow the idea first introduced in [38] and consider the value computed (in a distributed way) by the WSN as the result of a function that accepts as inputs the readings of a reasonable portion of the sensors the WSN is composed of.

In particular, we focus on the computation of the *MAX* of the readings performed by the sensors. Note that this is not a limitation, since the protocol can easily accommodate the computation of the MIN, as well as to verify whether sensors' readings are within a given range of values. In particular, note that this last capability is a building block for more complex query.

3.2. Network setup

Our protocol assumes a tree communication model where the root is the BS receiving data from the leaves which are the sensors. Intermediate sensors simply act as routers. Therefore, only the leaves send data. Extending the proposed solution to accommodate data transmission from intermediate sensors is straightforward. Note that the process used to form the tree hierarchy is outside the scope of this paper. Interested readers can refer to [7,2] for examples.

We also assume that the network is loosely time synchronized. Observe that loose time synchronization can be achieved both in a centralized and in a distributed way [19,36]. Hence, when the time-out expires, computation can be moved forward—sensors are required to send their contribution within the time-out. Further, to avoid message loss due to both shared media and high node density, a few MAC protocols have been proposed in the literature that cope with this issue [13].

Let n be the number of leaves and L_i, $1 \leqslant i \leqslant n$, denote the leaves. Each leaf L_i shares a unique, symmetric key K_i with the BS. Leaves are lightweight devices capable of computing hash functions (*e.g.* SHA-1 [34]). Intermediate sensors are also resource-constrained devices capable of computing only bitwise operations. Finally, the BS is a full-fledged computer.

Fig. 1 shows a simple scenario with a BS, two intermediate sensors (R_1 and R_2) and four leaves ($L_1,\ldots,L_4$). Note that, even though Fig. 1 represents a binary tree, our scheme is not linked to any specific tree structure.

3.3. Adversary model

We identify two *objectives* that an ADV has in thwarting a WSN:

- ADV could be interested in identifying the sensor(s) that satisfy the query issued by the BS. We assume that ADV has network wide eavesdropping capabilities.
- ADV could provide the BS with bogus data, so that the query-based decision process of the BS could be possibly disrupted or subverted towards incorrect decisions. This goal has been described in [38] as a *stealthy* attack, i.e., to cause the querier to accept a false data that is higher or lower than the true aggregate value. It is out of the scope of the paper to consider denial-of-service (DoS) attacks where the goal of ADV is to prevent the querier from getting any aggregation result at all. Furthermore, any maliciously induced extended loss of service is a detectable anomaly which will (eventually) signal the presence of ADV; subsequent protocols or manual intervention are assigned the task to resolve the problem.

Note that it is out of the scope of this paper to protect the confidentiality of the query result (that is, the MAX value): Our goal is to provide a protocol that could enforce location privacy and resilience. Indeed, in some applications event location (that is, "which"

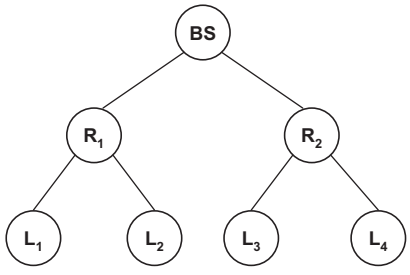

Fig. 1. Sample scenario.

sensor sensed a given event) could be more valuable than the content itself. However, for the sake of completeness, the solutions in [10,9,17] can be leveraged to address confidentiality issues in WSN. Finally, in this paper we consider an ADV able to compromise any sensor in the WSN. This adversary model has been adopted also in [16,5] to cite a few.

4. Our proposal

In the following we will detail our proposal. We will first provide a deterministic scheme that achieves the desired features, while in the following we will relax the proposal obtaining a more efficient, probabilistic solution. The trade-off is that, with a small probability (that is a design parameter), the query algorithm could return a value that is close to the MAX, but differs from it.

4.1. Deterministic solution

Each sensor (that is, a leaf in the WSN hierarchy) uses its sensing capabilities to acquire information from the environment. Sensed data is represented by an integer $S \in [0,\ldots,P]$, where P is the maximum integer value that can be read by a sensor. The binary representation of S is stored into an array A of d bits. It takes d rounds to send the complete d-bit sequence A from each leaf towards the BS.

We next detail the steps needed to transmit the bit located in the jth position ($j \in [0,\ldots,d-1]$) in sequence A (it represents a single round). These steps are executed until all d bits have been sent.

(1) QUERY. The BS generates a random value v and broadcasts it to all leaves.[1]
(2) MESSAGE GENERATION.
  (a) Upon receiving v, if it passes the authentication check, each leaf L_i computes a pseudo-random value $r_i = \mathscr{H}(v\|K_i)$, where $\mathscr{H}$ is a one-way hash function and $\|$ is the concatenation operator. If the authentication check fails, value v is simply discarded.
  (b) L_i computes $T_i = r_i + A[j]$.
  (c) L_i sends T_i up to its parent sensor.
(3) MESSAGE AGGREGATION. An intermediate sensor R (or the BS) receives messages from its w routers/leaves (children) and does the following:
  (a) Once all w expected messages T_w have been received, aggregate them by computing $T = \sum_w T_w$.
  (b) If R is not the BS, send T up to its parent sensor. Else, this is the final aggregated message.
(4) DATA EXTRACTION. From the final aggregated message T, the BS obtains how many leaves have transmitted $A[j] == 1$. This is done as follows:
  (a) Compute the pseudo-random value r_i linked to each leaf L_i in the current round j. Then, sum up all the n values: $\beta = \sum_n r_n$.
  (b) Compute $\alpha = T - \beta$. Value α represents how many leaves have sent "1" towards the BS in round j.
(5) NOTIFICATION. After the data extraction, the BS notifies to the leaves the following:
  - If $\alpha > 0$: All sensors that sent "0" ($A[j] == 0$) mark themselves as "inactive". It means that in all the following rounds they must send a zero as sensed value (an "inactive" sensor will keep this state until all d rounds have been executed). Sensors that sent "1" ($A[j] == 1$) mark themselves as "active". It means that they continue sending their sensed data as above described.
  - Else (this situation occurs when no one has sent "1"): No sensor is marked as "inactive". All "active" sensors will send their sensed data as usual in the next round $j+1$.

Let us assume that pseudo-random values r_i can reach a maximum value of Z. Hence, the total overhead experienced by links close to the BS is of $(1 + \lceil \log Z \rceil + \lceil \log n \rceil) \cdot \lceil \log P \rceil$ bits. With this solution, the BS is able to compute not only the value MAX, but also the number of MAX that are stored on leaves. However, in certain contexts, knowing the exact number of MAX provides the BS with redundant information. For instance, the BS could be interested in acquiring just the MAX value, while the number of sensors sensing the MAX value could be of no interest. Under this assumption, in the next section we provide a protocol satisfying these requirements with an average transmission overhead of only $(2 \cdot \lceil \log P \rceil)$ bits.

4.2. Probabilistic solution

4.2.1. Basic case

The protocol provided in the previous section on the one hand can be modified to reduce the produced message length. On the other hand, this modification implies some information loss. In particular, the new protocol enables the BS to know only the MAX value that is stored on leaves, while losing the information on the number of leaves that store this MAX value.

The modification is based on using the bitwise XOR operation ($\oplus$) instead of the addition operation. This change occurs in Step 2 and Step 3 (message generation and message aggregation respectively). According to that, leaf L_i generates at each jth round a pseudo-random value $r_i \in \{0,1\}$—for instance: $r_i = \oplus_{g=0}^{\ell-1} \mathscr{H}_g(v\|K_i)$. After that, L_i computes:

$$T_i = r_i \oplus A[j]$$

Then, L_i sends T_i up to its parent sensor. Later, any intermediate sensor simply performs the same operation on the received values and sends the result to its own parent in the tree hierarchy.

When the BS receives the aggregated data T, it can compute the pseudo-random value r_i linked to each leaf L_i in the current round j. Therefore, the BS is able to decrypt T:

$$\alpha = T \oplus \left(\oplus_{i=1}^{n} r_i\right) = \left[\oplus_{i=1}^{n} (r_i \oplus A_i[j])\right] \oplus \left(\oplus_{i=1}^{n} r_i\right) = \oplus_{i=1}^{n} A_i[j]$$

Value α represents whether the aggregated data at round j is zero or one. Then, the BS notifies the leaves in accordance with the Step 5 of the deterministic protocol.

4.2.2. Reducing the error probability

Note that in case that there is an even number of leaves sending "1" towards the root, the BS could be lead to take an incorrect decision. For instance, let us assume that, at round j, there are just two leaves which are going to send "1" (bit $A[j]$ is equal to one in both leaves). Following the protocol described in the previous section, at a certain intermediate sensor, the two bits will be XORed together and will be concealed. Therefore, when the BS will extract the data it will get $\alpha = 0$, inferring that there is no leaf sending "1" at round j. There are two facts regarding this situation:

- It can occur for any of the bits in the A sequence; but,
- it only occurs when the number of sensors sending "1" is even.

We refer to this error as a *false negative*. We next propose a method that reduces the probability for this error to occur; it is based on breaking the evenness of the signalling sensors.

[1] This query can be transmitted using a *broadcast authentication protocol* suitable for lightweight scenarios like WSN, using for instance solutions [16,6,27,33].

Assume that the number of sensors such is that $A_i[j] == 1$ is greater than zero. If we want to have the certainty that there is a $A_i[j] == 1$ signalled to the root at round j, we need an odd number k of leaves transmitting "1". To force (probabilistically) this case to occur, we randomize the contributions of each leave as follows: Each element that has to signal "1" decides with probability 1/2 whether to send either "0" or "1". In this way, if the number of contributors k is even, the number of contributors will be reduced to an odd number with a probability equal to 1/2.

Nevertheless, note that if k is odd, then it would have no sense to start with this randomization process. Therefore, it is better to start the first round without randomization since, if k is odd, that will be detected with probability 1. However, if the result is zero, that could be a false negative. Note that, on the average, if the number of sensors that have a "1" is even, two rounds of randomization will reveal that there is at least a sensor that is signalling a "1". Indeed, being the number of "1" even, the first round without randomization will fail by construction; however, one round of randomization fails with probability 1/2, hence we need two rounds of randomization on the average to detect a "1". On the average, the bit length of message routed by sensors close to the BS is $2 \cdot \log P$, which is considerably smaller than the $(1 + \log Z + \lfloor \log n \rfloor) \cdot \log P$ message length provided by the deterministic protocol.

Another situation occurs when all leaves send a zero value. In this case, even following the above protocol, all the results that are sent to the root will be extracted as a zero; that is correct. However, the BS cannot discern deterministically whether it received a zero because k is even, or because k is equal to zero. Fortunately, the BS can discern this situation with a certain probability. That is, we set a threshold value φ that indicates the maximum number of queries that, providing a result of zero, will make the BS to assume that all leaves are sending "0".

Note that the BS can incur in *false negative* type of error only. Indeed, if after the decryption phase the BS detects that at least one sensor sent a "1", than this event happened with probability 1—by construction, the protocol cannot experience *false positive*.

4.2.3. Accuracy of the received data

When $\alpha == 1$, there is no possibility of error: At least one of the sensors L_i is signalling that, in round j, $A_i[j] == 1$. However, when $\alpha == 0$, then two cases could have occurred:

- All sensors at round j have a zero in their $A[j]$ bit.
- There are an even number of sensors that are signalling that their $A[j]$ bit is "1".

When either of these two cases occurs, the BS queries the sensors that are signalling $A[j] == 1$ to randomize their contributions, as detailed in the previous section. Sensors are required to randomize their values a maximum of φ times—a design parameter. If each of the φ randomizations signals that the final aggregated bit is still a "0", the BS infers that all sensors at round j have a zero in their $A[j]$ bit. Note that this deduction is false (that is, the BS incurs in a false negative) with probability $(1/2)^{\varphi}$. Hence, the probability of a false negative decreases exponentially fast as the value φ increases. Nevertheless, a high φ also implies several protocol executions to guarantee the correctness of a single bit. If the same φ is applied to all the bits in A, an overall message length of $\varphi \cdot \log P + \log P$ bits is achieved—the second term of the sum is given by the non randomized round introduced in Section 4.2.2. However, this cost can be reduced by applying a different φ to each bit. This makes sense because not all the bits of the received sequence A have the same importance. Indeed, the bit situated in position j is more important than the one situated in position $j + 1$. As an example, assume that leaf L_i sensed the decimal value 7, and that this value is the maximum sensed by the WSN. This value has $A_i = \langle 1,1,1 \rangle$ as binary representation. If the protocol introduces a false negative in position $j = 0$, the BS will receive $A = \langle 0,1,1 \rangle$. Thus, the BS will reconstruct 3 as maximum value instead of 7. However, if the protocol introduces a false negative in position $j = 2$ ($A = \langle 1,1,0 \rangle$), the BS will reconstruct 6 as maximum value, that is very close to the correct maximum value 7. In general, if a false negative occurs in position j ($j = 0, \ldots, \log p - 1$; where position 0 is the leftmost significant bit), it will introduce an error of $2^{\log p - 1 - j}$.

Let φ_j the number of randomization set for the transmission of bit $A[j]$, where each φ_j is a design parameter. When selecting the φ_j values, the following constraints should be considered:

- Let Ψ the total overhead experienced by links close to the BS. The final aggregated message has a bit-length of $\Psi + \log P$ where $\Psi = \sum_{j=0}^{\log P} \varphi_j$.
- The optimal value for each φ_j is obtained from a trade-off between the message length of the protocol and its accuracy. A high Ψ (which represents a high overhead) guarantees high accuracy of the received data. Our objective is to reduce the Ψ to the minimum while preserving an acceptable degree of accuracy. Hence, given a fixed Ψ, the optimal φ_j selection is the solution of the following minimization problem:

$$\min \left(\sum_{j=0}^{\log P - 1} (1/2)^{\varphi_j} \cdot 2^{\log P - 1 - j} \right) \tag{1}$$

Here, $(1/2)^{\varphi_j}$ denotes the probability of receiving an erroneous jth bit and $2^{\log P - 1 - j}$ represents the error introduced by the receiving a bit that is a false negative. Since this error must be as low as possible, the solution of this minimization problem gives the best choice for each φ_j.

4.2.4. Message flow of the probabilistic protocol

In the following we detail the steps of our probabilistic protocol via an example.

Fig. 2 shows the message flow generated between the BS and the sensors during a protocol execution. This figure represents a simple scenario with a BS, two intermediate sensors (R_1 and R_2) and four leaves ($L_1, \ldots, L_4$). In Fig. 2a the BS broadcasts a query to all leaves (Step 1 in the protocol execution). In Fig. 2b, message (1) sent by L_1 corresponds to T_1 while message (2) sent by L_2 corresponds to T_2 (Step 2 in the protocol execution). Sensor R_1 constructs message (3), that corresponds to $T_{R_1} = T_1 \oplus T_2$, by aggregating messages (1) and (2) (Step 3 in the protocol execution). The same process occurs in the subtree rooted at R_2. The latter sensor constructs message (6) by aggregating messages (4) and (5), that correspond to T_3 and T_4, respectively. Eventually, the BS aggregates messages (3) and (6) to get the final aggregated message. After that, the BS extracts the symbols transmitted by the leaves (Step 4 in the protocol execution).

Now, let us assume that L_1 is transmitting $A_1 = \langle 1,1 \rangle$, L_2 is transmitting $A_2 = \langle 1,0 \rangle$ and both L_3, L_4 are transmitting $A_3 = A_4 = \langle 0,0 \rangle$. In the first round ($j = 0$) the BS extracts $T = 0$. However, this is a *false negative*, since there are leaves that sent a non zero value (L_1 and L_2). As explained in Section 4.2.2, this false negative occurred because an even number of sensors signalled that their $A[0]$ bit is "1". We will use again Fig. 2 to explain how the system overcomes this erroneous situation.

Let us assume that $\varphi_0 = 2$ and $\varphi_1 = 1$ (this implies $\Psi = 3$). When $j = 0$ the BS asks the leaves three times (the first query, that is always non randomized, plus two randomized queries) before accepting that all leaves are sending $A_i[0] = 0$. Assume that after the second query (represented by Fig. 2a) linked to $j = 0$ (we are still in the first round) L_1 and L_2 send 0 and 1 respectively (which

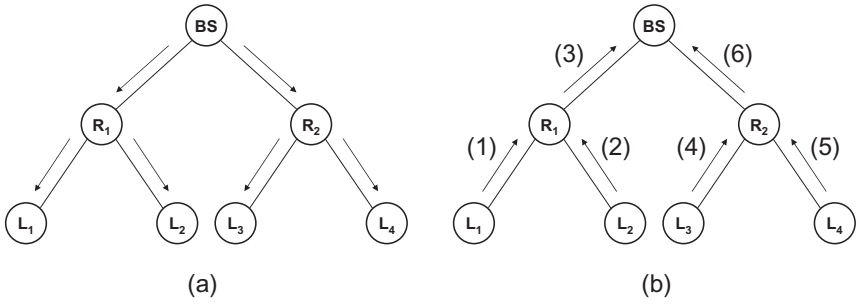

Fig. 2. Message flow between the BS and the leaves.

correspond to messages (1) and (2) in Fig. 2b). Note that L_3 and L_4 transmit 0 like in the first query (messages (3) and (4) in Fig. 2b). As a result, the BS extracts $T = 1$ as a final value for the first round ($j = 0$). This is the correct value. Before launching the second round of the protocol, the BS notifies (Step 5 in the protocol execution) all leaves that each sensor that has signalled $A[0] == 0$ has to remain "inactive". Such a sensor must send a zero as sensed value in the following rounds. As a result, L_3 and L_4 mark themselves as "inactive". Note that once a sensor marks itself as inactive, it does not change its state for the remaining rounds of the protocol.

In the second round ($j = 1$), only L_1 is signalling that its $A_1[1]$ bit is "1"—a false negative cannot occur. Consequently, the BS extracts $T = 1$ as a final value for the second round ($j = 1$). Eventually, the BS reconstructs the maximum value sensed by the leaves of the WSN as $A = \langle 1, 1 \rangle$.

5. Security analysis

5.1. Resilience to passive attacks

Let us consider an ADV eavesdropping all messages sent between the sensors and the BS. Our protocol prevents this ADV from knowing which reading has sent each sensor.

Assume that this intruder focuses on a given message T_i sent by leaf L_i. From T_i, the attacker is able to determine:

$$T_i = r_i \oplus A[j]$$

In this equation, $A[j]$ corresponds to the bit that the adversary wants to know and r_i is a pseudo-random bit computed as follows:

$$r_i = \oplus_{g=0}^{\ell-1} \mathcal{H}_g(v \| K_i)$$

That is, r_i is given by the parity computed over the hash of the challenge v concatenated with the key K_i—where $\mathcal{H}_g(v\|K_i)$ is the gth bit of the computed hash, assumed to be over ℓ bits. Note that the challenge v is known by every sensor in the network (and possibly by ADV), but K_i is a symmetric key shared between L_i and the BS only. Hence, since ADV cannot compute r_i, it cannot retrieve $A[j]$.

5.2. Resilience to active attacks

Let us consider an ADV that: Has compromised a set of sensors ω; and, its aim is to prevent the BS from obtaining the correct MAX value.

If the MAX value is only sensed by sensors in ω, ADV can modify the sensed value to any value different from MAX, and ADV wins. Moreover, ADV can also provide a bogus MAX value to the BS, sending via a corrupted sensor a "sensed" value that is always set to the ceiling of the sensing range. However, note that the BS could perform some audit activity—when it realizes that some sensors are sending outliers values—eventually identifying corrupted sensors. In this sense, this attack is out of the scope of this paper. However, in the following we sketch a possible countermeasure: intermediate sensors can store some bits regarding received messages so that it could be possible for the BS, when the audit takes place, to trace back the message originator. Note that this countermeasure does not introduce a significant overhead—details and implementation are left for further study.

We assume that ADV wants to maximize its chances to be stealthy. Therefore, in the following we analyze the more interesting case where the MAX has been sensed by a non-empty set of non-compromised sensors ($\mathcal{S}$), and ADV attempts to disrupt the computation, that is having the BS to receive a different value, while minimizing its chances to be detected. Moreover, we also assume that ADV knows the value MAX that is going to be sent towards the BS. The only hypothesis required by our protocol to enforce resilience is that the set of non-compromised sensors that have sensed the current MAX is non empty—that is, $\mathcal{S} \neq \emptyset$.

Under these assumptions and following the proposed protocol, ADV can attempt to disrupt the computation of the MAX in two ways: Either causing a false negative, or causing a false positive.

As for the false negative, ADV has to contribute to the protocol (via the corrupted sensors) in such a way that the number of sensors signalling "1" is even. Note that we can discriminate two further cases: First, ADV is aware of the number of the leaves that sensed MAX; second, it is not. However, as shown in the following, in both cases ADV has no better choice than to randomly decide whether to send an odd number of "1s".

Indeed, each sensor in $\mathcal{S}$ that is "active" and that has to send a "1", will randomly decide whether to send "1" or "0". The probability that the number of sensors that will send a "1" is odd is exactly 1/2 (conversely, the probability that the number of sensors that will send a "0" is even is 1/2 as well). Further, note that this is true regardless of the number of sensors in $\mathcal{S}$ that, at each round, are still marked "active". Hence, ADV has no deterministic advantage in contributing (via the corrupted sensors) to the aggregated value with an odd number of "1s". Indeed, for ADV to cause a false negative, it has to guess the correct value to send for each one of the φ_i rounds run by the sensors to send their readings to the BS (see Section 4.2.3).

As a result, an attacker can perform an active attack on the ith bit of the binary representation of the MAX value with a success probability of $(1/2)^{\varphi_i}$ only.

As for the *false positive*, we start noticing that the proposed protocol cannot produce it. Indeed, no sensor that has to transmit a "0" will send a "1". Let us assume that an active ADV is present and no active sensor in $\mathcal{S}$ has to transmit a "1". ADV could introduce an odd number of "1s" via the corrupted sensors—note that for ADV to reach this goal, in this case just a corrupted sensor would suffice. However, if ADV signals an odd number of "1s", that would mean that from that round on, all the sensors in $\mathcal{S}$ will mark themselves as "inactive"; indeed ADV is sending a value MAX′, that satisfies MAX′ > MAX. The countermeasure would be the following one: Once the BS detects the fact that sensed data is incorrectly

reported, the BS can start a detection phase backtracking the intermediate sensors till the leaves. This way, the leave(s) that reported the incorrect value could be tracked and subsequently evicted from the WSN. However, detection and eviction mechanisms are outside the scope of this paper—interested readers could refer to [3,15].

6. Simulations and discussion

We have simulated our protocol to observe its performance. More specifically, we decided to check the success probability of a protocol execution in different scenarios. In these simulations, we consider that the protocol succeeds when the BS extracts the exact MAX value stored on leaves. It means that we require perfect accuracy in all the bits of the received data. That is, the same φ is linked to each bit. As mentioned in Section 4.2.2, φ is a threshold value that indicates the maximum number of queries that, providing a result of zero, will make the BS to assume that all leaves are sending "0".

The proposed scheme was tested for four scenarios with different requirements regarding the length of the sensed data. In particular, we considered the MAX value over P to require 8, 16, 24 and 32 bits to be represented.

In each scenario, the sensors of the network formed a binary tree (although our scheme is not linked to a fixed tree structure). For each scenario, networks formed by 128, 256, 512 and 1024 leaves (sensors sending their sensed data towards the BS) were considered. The φ values used in the simulations ranged from 0 to 12, with an incremental step of two. Note that if $\varphi = 0$, the proposed method for reducing the error probability (Section 4.2.2) is not used. For each combination between the number of leaves and φ, 997 tests were run and the average results were computed. Given a certain success probability s (in %), 997 tests assures that interval $[s - 5\%, s + 5\%]$ is a 95% confidence interval for parameter s.

The results for the four scenarios considered are shown in Tables 1–4. The content of these tables is commented below.

6.1. Simulation results

From the results in Tables 1–4, it can be observed that an increase in the number of leaves sending data does not produce a significant decrease in the success probability. Also, the results show that the better success probability is obtained in scenarios where the length of the data to be transmitted is smaller. This follows from the following consideration: Let us assume that the probability of receiving correctly a single bit is $1/y$; the probability p to receive a correct sequence of d bits is $p = (1/y)^d$.

Regarding the method proposed to reduce the error probability, if this method is not used (it happens when $\varphi = 0$), the protocol does not work properly in any scenario (the best result is a 1.8% of success probability). However, with just $\varphi = 2$ the protocol performance improves dramatically. Note that the message length is $\varphi \cdot \log P + \log P$ bits.

For $\varphi = 10$ and the number of leaves ranging from 128 to 1024, the protocol achieves a success probability above the 99% for all four tested data lengths (8, 16, 24 and 32 bits). This behaviour proves that the φ value provides an exponential gain in the accuracy of the provided results.

Finally, note that the number of transmitted bits towards the BS depends only on the value φ, while it is not related to the number of sensors in the network. Moreover, the error probability is only weakly correlated to the number of sensors in the WSN—a correlation that becomes being not relevant for small values of φ already. Hence, our proposal enjoys excellent scalability.

Table 1
Success probability when transmitting 8 bits of sensed data ($P = 255$).

# leaves	Success probability (%)						
	$\varphi = 0$	$\varphi = 2$	$\varphi = 4$	$\varphi = 6$	$\varphi = 8$	$\varphi = 10$	$\varphi = 12$
128	1.8	42.3	82.1	98.3	99.3	100.0	100.0
256	1.0	39.9	78.1	95.6	98.6	99.3	100.0
512	0.5	34.6	78.4	94.3	99.0	99.7	100.0
1024	0.6	34.0	77.5	92.9	98.4	99.6	100.0

Table 2
Success probability when transmitting 16 bits of sensed data ($P = 65{,}535$).

# leaves	Success probability (%)						
	$\varphi = 0$	$\varphi = 2$	$\varphi = 4$	$\varphi = 6$	$\varphi = 8$	$\varphi = 10$	$\varphi = 12$
128	1.4	41.8	83.1	97.1	98.9	100.0	100.0
256	0.8	36.9	79.6	94.3	99.1	99.7	100.0
512	0.5	33.1	76.1	94.1	98.6	100.0	100.0
1024	0.5	33.2	74.4	91.6	98.1	99.2	100.0

Table 3
Success probability when transmitting 24 bits of sensed data ($P = 2^{24} - 1$).

# leaves	Success probability (%)						
	$\varphi = 0$	$\varphi = 2$	$\varphi = 4$	$\varphi = 6$	$\varphi = 8$	$\varphi = 10$	$\varphi = 12$
128	1.2	39.5	80.2	95.1	98.8	99.3	100.0
256	0.9	37.4	77.1	94.4	98.1	99.0	100.0
512	0.3	31.6	75.6	93.0	97.7	99.4	99.9
1024	0.4	28.4	73.9	93.3	98.2	99.2	99.9

Table 4
Success probability when transmitting 32 bits of sensed data ($P = 2^{32} - 1$).

# leaves	Success probability (%)						
	$\varphi = 0$	$\varphi = 2$	$\varphi = 4$	$\varphi = 6$	$\varphi = 8$	$\varphi = 10$	$\varphi = 12$
128	0.9	36.8	79.4	95.3	98.0	99.5	100.0
256	0.7	29.8	76.9	93.5	98.6	99.3	100.0
512	0.5	28.2	74.6	92.6	97.7	99.7	99.8
1024	0.5	26.1	74.8	92.7	97.7	99.7	99.9

6.2. Resilience to an active adversary

In this subsection we report on the simulations performed to assess the resilience of the proposed protocol against an active adversary. ADV is aiming at providing bogus data to the BS, while remaining stealthy, in accordance to the adversary model described in Section 3.3 and discussed in Section 5.2.

The strategy of the attacker has been highlighted in previous section; that is, at each round ADV decides whether to send a 1 or a 0 with probability 1/2. Note again that ADV needs to compromise just one sensor to start applying this strategy. This could seem to provide ADV with a strong advantage: With just one compromised sensor, it could attempt to feed the BS with bogus data. However it should be noted that, even if ADV has compromised more than one sensor, it cannot do anything better—that is, its payoff does not scale at all when compromising more sensors; and, as the following result confirm, the payoff received by ADV is almost negligible.

The same four scenarios analyzed in the previous section are simulated again, with the only difference that the incremental step the variable φ is subject to is four. The results for the four scenarios considered are shown in Table 5–8. It can be observed that the success probability to obtain the correct MAX value decreases just slightly under an active attack in comparison with the case where no ADV is present (the difference is in any case no grater than 0.8%). In particular, for $\varphi \geqslant 8$ the payoff of the attack strategy

Table 5
Success probability when transmitting 8 bits of sensed data under an active attack.

# leaves	Success probability (%)			
	$\varphi = 0$	$\varphi = 4$	$\varphi = 8$	$\varphi = 12$
128	1.7	81.4	98.8	99.9
256	0.6	76.7	98.9	99.9
512	0.3	77.2	97.8	99.7
1024	0.7	77.3	97.7	99.8

Table 6
Success probability when transmitting 16 bits of sensed data under an active attack.

# leaves	Success probability (%)			
	$\varphi = 0$	$\varphi = 4$	$\varphi = 8$	$\varphi = 12$
128	0.5	79.8	97.6	100.0
256	0.4	76.6	98.9	99.6
512	0.0	73.2	97.7	99.7
1024	0.0	72.6	98.3	99.6

Table 7
Success probability when transmitting 24 bits of sensed data under an active attack.

# leaves	Success probability (%)			
	$\varphi = 0$	$\varphi = 4$	$\varphi = 8$	$\varphi = 12$
128	0.6	76.2	98.6	100.0
256	0.2	74.1	98.6	100.0
512	0.2	71.8	97.3	99.8
1024	0.0	70.2	97.2	99.7

Table 8
Success probability when transmitting 32 bits of sensed data under an active attack

# leaves	Success probability (%)			
	$\varphi = 0$	$\varphi = 4$	$\varphi = 8$	$\varphi = 12$
128	0.6	77.1	98.6	100.0
256	0.6	73.8	98.5	99.7
512	0.2	71.3	97.2	100.0
1024	0.0	71.1	97.2	99.7

becomes negligible. That is, the proposed protocol thwarts the capabilities of an active ADV to provide the BS with bogus data.

6.3. Implementation issues

Note that we have assumed that the BS queries the leaves in rounds, where in each round the leaves send up in the hierarchy just one bit. Hence, to transmit an overall of $\varphi \cdot \log P + \log P$ bits (for each bit to transmit, the leaves first send the non randomized value, while in the φ later rounds, they send the randomized reply to the query—if needed), that value would also be the maximum number of required rounds. However, note that it could be possible to send all the φ bits of the randomized replies related to a single bit of the sensed value, in just one message. Hence, the overall number of rounds required could be reduced to $\log P + 1$, where just φ bits of overhead are required per-round—and just one bit for the first round, where randomization is not used. Further, note that data transmission in real WSN is per-packet; for example, the default TinyOS active message leaves 29 bytes for application payload data. Hence, one round could require just 1 byte overhead (for instance, selecting $\varphi = 8$), while the first round would require just one bit overhead—this way, the protocol could achieve a success probability greater than 97%, as shown in Tables 1–4. Moreover, assuming that there is more than a single application running on motes (indeed, querying could be thought just as an audit activity), the byte overhead could be piggybacked within the payload of the main application.

Finally, we present a synthetic comparison with a similar protocol that provides the same properties. Bearing in mind that our solution discloses no information to an attacker—other than the information it can gather form the sensors it is in control of—an equivalent solution to our protocol from the privacy point of view would be to have each node to encrypt the sensed value with a key shared with the BS only, and to send this value upwards in the tree. Assuming a length leaf-BS of $\log n$, our solution requires transmitting an overall of $2n(\varphi \log P + \log P)$ bits, while the privacy equivalent solution would require $n \log n \log P$ bits. While these two solutions require a roughly equal number of bits to be sent, note that the latter one incurs into the message implosion phenomenon [35], that makes this solution not viable, while our proposal is immune from the implosion curse, being the transmission evenly spread among all the wireless links.

As for the accuracy of the computed result, we dealt with this issue in Section 4.2.3, that lead to Eq. (1).

7. Concluding remarks

In this paper, we have addressed the problem of preserving the location privacy of sensors of a WSN when replying to a query broadcast by the BS. First, we have provided a deterministic solution that reveals to the BS both the MAX value sensed by the WSN and the number of sensors that sensed that value, while preserving the privacy of the sensors that sent that reading. Later, we have relaxed the problem, and introduced a probabilistic protocol that only notifies the BS of the MAX value stored by leaves. This second solution is highly efficient, trading off this improvement in efficiency with a possible (tunable) loss in the accuracy of the query result.

The proposed protocol enjoys several features: It enforces location privacy of the sensors participating in the protocol; it is resilient to an active ADV aiming at disrupting the computation; it is completely customizable in its security and accuracy parameters; it is scalable; and, the introduced (low) overhead could be nicely amortized over the available message payload. Finally, thorough analysis and extensive simulations validate our findings.

As for future work, we are extending the proposed protocol to support other functions, such as range query and top k-query.

Acknowledgments

The authors are solely responsible for the views expressed in this paper, which do not necessarily reflect the position of UNESCO nor commit that organization. This work was partly supported by the Spanish Ministry of Education through projects TSI2007-65406-C03-01 "E-AEGIS" and CONSOLIDER CSD2007-00004 "ARES", and by the Government of Catalonia under Grant 2009 SGR 1135.

The authors thank the anonymous reviewers for their comments that helped improving the paper.

References

[1] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, Erdal Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393–422.
[2] T. Banerjee, K. Chowdhury, D.P. Agrawal, Tree based data aggregation in sensor networks using polynomial regression, in: 8th International Conference on Information Fusion, July 2005, pp. 469–477.
[3] Levente Buttyán, Péter Schaffer, István Vajda, Ranbar: Ransac-based resilient aggregation in sensor networks, in: SASN'06: Proceedings of the Fourth ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM, New York, NY, USA, 2006, pp. 83–90.

[4] Bogdan Carbunar, Yang Yu, Weidong Shi, Michael Pearce, Venu Vasudevan, Query privacy in wireless sensor networks, in: SECON, 2007, pp. 203–212.
[5] Claude Castelluccia, Aldar C.-F. Chan, Einar Mykletun, Gene Tsudik, Efficient and provably secure aggregation of encrypted data in wireless sensor networks, ACM Transactions on Sensor Networks 5 (3) (2009) 1–36.
[6] Shang-Ming Chang, Shiuhpyng Shieh, Warren W. Lin, Chih-Ming Hsieh, An efficient broadcast authentication scheme in wireless sensor networks, in: ASIACCS'06: Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security, ACM, New York, NY, USA, 2006, pp. 311–320.
[7] Santashil Pal Chaudhuri, S. Du, A.K. Saha, D.B. Johnson, Treecast: a stateless addressing and routing architecture for sensor networks, in: Proceedings of the 18th International Parallel and Distributed Processing Symposium, April 2004, p. 221.
[8] William Conner, Tarek F. Abdelzaher, Klara Nahrstedt, Using data aggregation to prevent traffic analysis in wireless sensor networks, in: DCOSS, 2006, pp. 202–217.
[9] Mauro Conti, Roberto Di Pietro, Luigi Vincenzo Mancini, Ecce: enhanced cooperative channel establishment for secure pair-wise communication in wireless sensor networks, Ad Hoc Networks 5 (1) (2007) 49–62.
[10] Mauro Conti, Lei Zhang, Sankardas Roy, Roberto Di Pietro, Sushil Jajodia, Luigi V. Mancini, Privacy-preserving robust data aggregation in wireless sensor networks, Security and Communication Networks 2 (2) (2009) 195–213.
[11] Emiliano De Cristofaro, X. Ding, Gene Tsudik, Privacy-preserving querying in sensor networks, in: Proceedings of the IEEE 18th International Conference on Computer Communications and Networks (ICCCN'09), 2009, pp. 1–6.
[12] Emiliano De Cristofaro, Stanislaw Jarecki, Jihye Kim, Gene Tsudik, Privacy-preserving policy-based information transfer, in: Privacy Enhancing Technologies, 2009, pp. 164–184.
[13] I. Demirkol, C. Ersoy, F. Alagoz, MAC protocols for wireless sensor networks: a survey, IEEE Communications Magazine 44 (4) (2006) 115–121.
[14] Jing Deng, Richard Han, Shivakant Mishra, Countermeasures against traffic analysis attacks in wireless sensor networks, in: SECURECOMM'05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, IEEE Computer Society, Washington, DC, USA, 2005, pp. 113–126.
[15] Roberto Di Pietro, Luigi V. Mancini, Sushil Jajodia, Secure selective exclusion in ad hoc wireless network, in: SEC'02: Proceedings of the IFIP TC11 17th International Conference on Information Security, Kluwer, B.V., Deventer, The Netherlands, 2002, pp. 423–434.
[16] Roberto Di Pietro, Luigi V. Mancini, Alessandro Mei, Energy efficient node-to-node authentication and communication confidentiality in wireless sensor networks, Wireless Networks 12 (6) (2006) 709–721.
[17] Roberto Di Pietro, Luigi V. Mancini, Alessandro Mei, Alessandro Panconesi, Jaikumar Radhakrishnan, Redoubtable sensor networks, ACM Transactions on Information and System Security 11 (3) (2008).
[18] Roberto Di Pietro, Pietro Michiardi, Refik Molva, Confidentiality and integrity for data aggregation in WSN using peer monitoring, Security and Communication Networks 2 (2) (2009) 181–194.
[19] Jeremy Elson, Deborah Estrin, Time synchronization for wireless sensor networks, in: IPDPS'01, 2001, pp. 1965–1970.
[20] Xinwen Fu, Bryan Graham, Riccardo Bettati, Wei Zhao, Active traffic analysis attacks and countermeasures, in: ICCNMC'03: Proceedings of the 2003 International Conference on Computer Networks and Mobile Computing, IEEE Computer Society, Washington, DC, USA, 2003, p. 31.
[21] G.W. Hart, Residential energy monitoring and computerized surveillance via utility power flows, IEEE Technology and Society Magazine 8 (2) (1989) 12–16.
[22] Wenbo He, Xue Liu, Hoang Nguyen, Klara Nahrstedt, Tarek F. Abdelzaher, PDA: privacy-preserving data aggregation in wireless sensor networks, in: INFOCOM, 2007, pp. 2045–2053.
[23] James Horey, Michael M. Groat, Stephanie Forrest, Fernando Esponda, Anonymous data collection in sensor networks, in: MOBIQUITOUS'07: Proceedings of the 2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking& Services (MobiQuitous), IEEE Computer Society, Washington, DC, USA, 2007, pp. 1–8.
[24] Ying Jian, Shigang Chen, Zhan Zhang, Liang Zhang, Protecting receiver-location privacy in wireless sensor networks, in: INFOCOM, 2007, pp. 1955–1963.
[25] Pandurang Kamat, Yanyong Zhang, Wade Trappe, Celal Ozturk, Enhancing source-location privacy in sensor network routing, in: ICDCS'05: Proceedings of the 25th IEEE International Conference on Distributed Computing Systems, IEEE Computer Society, Washington, DC, USA, 2005, pp. 599–608.
[26] Yun Li, Jian Ren, Preserving source-location privacy in wireless sensor networks, in: SECON'09: Proceedings of the 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, IEEE Press, Piscataway, NJ, USA, 2009, pp. 493–501.
[27] Donggang Liu, Peng Ning, Multilevel micro-tesla: broadcast authntication for distributed sensor networks, ACM Transactions in Embedded Computing Systems 3 (4) (2004) 800–836.
[28] K. Mehta, Donggang Liu, M. Wright, Location privacy in sensor networks against a global eavesdropper, in: ICNP'07, 2007, pp. 314–323.
[29] Satyajayant Misra, Guoliang Xue, Efficient anonymity schemes for clustered wireless sensor networks, The International Journal of Sensor Networks 1 (1/2) (2006) 50–63.
[30] Stefano Ortolani, Mauro Conti, Bruno Crispo, Roberto Di Pietro, Event handoff unobservability in wsn, in: Proceedings of the 2010 IFIP Open Research Problems in Network Security Conference (iNetSec 2010), in press.
[31] Yi Ouyang, Zhengyi Le, Donggang Liu, James Ford, Fillia Makedon, Source location privacy against laptop-class attacks in sensor networks, in: SecureComm'08: Proceedings of the 4th International Conference on Security and Privacy in Communication Networks, ACM, New York, NY, USA, 2008, pp. 1–10.
[32] Celal Ozturk, Yanyong Zhang, Wade Trappe, Source-location privacy in energy-constrained sensor network routing, in: SASN'04: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM, New York, NY, USA, 2004, pp. 88–93.
[33] Adrian Perrig, Robert Szewczyk, J.D. Tygar, Victor Wen, David E. Culler, SPINS: security protocols for sensor networks, Wireless Networks 8 (5) (2002) 521–534.
[34] Federal Information Processing Standards Publications. Secure Hash Standard (SHS), Technical Report FIPS-180-1, National Institute of Standards and Technology, October 1995.
[35] B. Quinn, K. Almeroth, IP multicast applications: challenges and solutions, rfc 3170, 2001.
[36] David Sanchez, Secure, accurate and precise time synchronization for wireless sensor networks, in: Q2SWinet'07: Proceedings of the 3rd ACM Workshop on QoS and Security for Wireless and Mobile Networks, 2007, pp. 105–112.
[37] Latanya Sweeney, k-anonymity: a model for protecting privacy, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems 10 (5) (2002) 557–570.
[38] David Wagner, Resilient aggregation in sensor networks, in: SASN'04: Proceedings of the 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks, ACM, New York, NY, USA, 2004, pp. 78–87.
[39] Tilman Wolf, Sumi Choi, Aggregated hierarchical multicast – a many-to-many communication paradigm using programmable networks, IEEE Transactions on Systems, Man, and Cybernetics, Part C 33 (3) (2003) 358–369.
[40] Yong Xi, L. Schwiebert, Weisong Shi, Preserving source location privacy in monitoring-based wireless sensor networks, in: 20th International Parallel and Distributed Processing Symposium (IPDPS 2006), April 2006, 8 pp.
[41] Yi Yang, Min Shao, Sencun Zhu, Bhuvan Urgaonkar, Guohong Cao, Towards event source unobservability with minimum network traffic in sensor networks, in: WiSec'08: Proceedings of the First ACM Conference on Wireless Network Security, ACM, New York, NY, USA, 2008, pp. 77–88.
[42] Wensheng Zhang, Chuang Wang, Taiming Feng, GP^2S: generic privacy-preservation solutions for approximate aggregation of sensor data (concise contribution), in: PERCOM'08: Proceedings of the 2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications, IEEE Computer Society, Washington, DC, USA, 2008, pp. 179–184.