Probabilistic receiver-location privacy protection in wireless sensor networks

Ruben Rios[a], Jorge Cuellar[b], Javier Lopez[a]

[a] *Universidad de Málaga, NICS Lab, Campus de Teatinos, 29071 Málaga, Spain*
[b] *Siemens AG, Otto-Hahn-Ring 6, 81739 Munich, Germany*

Abstract

Wireless sensor networks (WSNs) are continually exposed to many types of attacks. Among these, the attacks targeted at the base station are the most devastating ones since this essential device processes and analyses all traffic generated in the network. Moreover, this feature can be exploited by a passive adversary to determine its location based on traffic analysis. This receiver-location privacy problem can be reduced by altering the traffic pattern of the network but the adversary may still be able to reach the base station if he gains access to the routing tables of a number of sensor nodes. In this paper we present HISP-NC (Homogenous Injection for Sink Privacy with Node Compromise protection), a receiver-location privacy solution that consists of two complementary schemes which protect the location of the base station in the presence of traffic analysis and node compromise attacks. The HISP-NC data transmission protocol prevents traffic analysis by probabilistically hiding the flow of real traffic with moderate amounts of fake traffic. Moreover, HISP-NC includes a perturbation mechanism that modifies the routing tables of the nodes to introduce some level of uncertainty in attackers capable of retrieving the routing information from the nodes. Our scheme is validated both analytically and experimentally through extensive simulations.

Keywords: Wireless sensor networks, security, privacy, traffic analysis, node capture

Email addresses: ruben@lcc.uma.es (Ruben Rios), jorge.cuellar@siemens.com (Jorge Cuellar), jlm@lcc.uma.es (Javier Lopez)

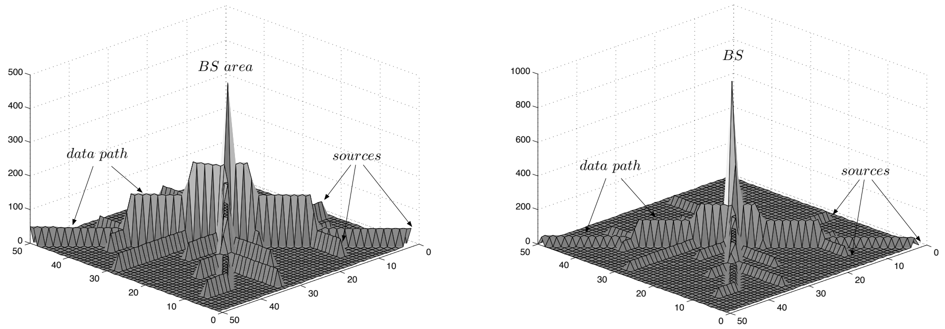
1. Introduction

Wireless sensor networks (WSNs) [30] are highly distributed networks comprising two types of devices, namely, the sensor nodes and the base station. The sensor nodes are small battery-powered computers, which have the ability to measure the physical phenomena (e.g., temperature, vibration, radioactivity) occurring in their vicinity and to wirelessly communicate with other devices nearby. The base station is a resourceful wireless-enabled device in charge of gathering the data coming from different sources and processing them in order to gain insight about the phenomena being monitored.

Due to the number of sensors they can incorporate, these networks are extremely versatile, which makes them suitable for countless application scenarios where sensor nodes are unobtrusively embedded into systems for monitoring, tracking and surveillance operations. Many of these applications are extremely sensitive and thus security and privacy become essential properties [27]. Extensive work has been done on the protection of sensor networks from the physical to the application layer but privacy preservation has only recently drawn the attention of the research community due to the imminent adoption of this technology in scenarios involving businesses, individuals and valuables assets.

Privacy threats in WSNs can be categorised as content-oriented and context-oriented [19]. Content-oriented privacy focuses on safeguarding the actual data sensed by the nodes [32] and the queries issued to the network [10]. On the other hand, context-oriented privacy refers to the protection of the metadata associated with the measurement and transmission of data. These data include, among other pieces of information [20], the time at which sensitive information is collected and the location of the nodes involved in the communication.

Similarly, there are two main types of location privacy problems affecting sensor networks: source- and receiver-location privacy. The former is concerned with hiding the area where a particular phenomenon is detected while the later is intended to prevent disclosing the location of the base station. For example, in a military scenario, sensor nodes may be deployed to monitor the troops and vehicles belonging to a military force for a better management and control. While moving on the battlefield, the sensors nodes transmit messages which are forwarded to the base station to inform about the units observed and their whereabouts. Even when secure cryptographic algorithms are used to protect the confidentiality and integrity of the communications,



(a) Number of packets sent

(b) Number of packets received

Figure 1: Single-path Routing Protocol

the mere presence of messages in the network reveal sensitive information to the enemy: there are troops somewhere in the field. With little effort and not overly sophisticated devices (e.g., a directional antenna) the attacker can deduce more information from the packets he observes and eventually traceback to the original data source. Similarly, the attacker could reach the base station and take control of the network or even render it useless by destroying this critical device. Both source- and receiver-location privacy are important properties but the latter is essential for the integrity and survivability of the network. Besides its importance for the physical protection of the network, the location of the base station is strategically critical because it is most likely housed in a highly-relevant facility. In the aforementioned military scenario, learning the location of the base station gives the attacker an important advantage over the enemy as it is likely housed at military base.

These privacy problems are extensible to any application scenario because they are caused by the singular communication model of WSNs. See in Figure 1 a typical WSN consisting of 50×50 nodes where 15 nodes are reporting event data using a shortest-path routing protocol. Although this is the most suitable configuration for preserving the limited energy budget of sensor nodes because it minimises the number of nodes involved in the communication process, it also produces very pronounced traffic patterns that reveal the location of both the source nodes and the base station. Ideally, the number of transmissions of each of the nodes should be similar in order to provide an adequate privacy protection level, however this implies a significant energy waste that negatively impacts the lifetime of the network. As a result, a number of techniques [8, 14, 31] have struggled to randomise the traffic pattern while preserving the limited energy budget of the nodes.

Besides performing traffic analysis attacks, an adversary can exploit the fact that each sensor node stores a routing table to allow the delivery of data to the base station. Normally, the routing tables contain information regarding the distance to or location of the base station. This information can be used by the attacker to effectively reach the base station after inspecting very few routing tables, thus rendering useless the efforts made by the network in deploying anti-traffic analysis mechanisms. Notwithstanding, none of the existing solutions in the literature of receiver-location privacy take this serious threat into consideration. This paper addresses, for the first time, both problems in a single solution, the HISP-NC (Homogenous Injection for Sink Privacy with Node Compromise protection) protocol. On the one hand, HISP-NC data transmission protocol hides the flow of real messages by introducing controlled amounts of fake traffic to locally homogenise the number of packets being forwarded from a sensor node to its neighbours. On the other hand, HISP-NC perturbation scheme modifies the routing tables of the nodes to reduce the risk of node capture attacks while ensuring that data packets eventually reach the base station.

The rest of the paper is organised as follows. Section 2 compares this work with previous location privacy solutions in the area of WSNs. Section 3 describes the network and threat models as well as the main assumptions applicable to the rest of the paper. A detailed description of the HISP-NC data transmission and routing tables perturbation schemes are presented in Section 4 and 5, respectively. Then, Section 6 evaluates and analyses the potential limitations of our solution with respect to the traffic overhead and delivery time of data packets. Next, Section 7 presents a discussion and evaluates the privacy protection level achieved by the HISP-NC scheme under different types of attacks. Finally, Section 8 provides some concluding remarks and outlines new directions for future work.

2. Related Work

This section compares HISP-NC with existing location privacy solutions in WSNs. For a deeper analysis and classification of location privacy solutions in WSNs refer to [22].

2.1. Source-Location Privacy

The source-location privacy problem was introduced in [19]. This work proposes the Phantom Routing protocol to counter adversaries tracing back packets to the source node. This protocol sends each message on a random or directed walk to a phantom source, which finally forwards the packet to the sink using a flooding-based or a single-path routing scheme. In this way, each packet appears to have originated from a different data source. This protocol presents several drawbacks especially in the walking phase, which tends to stay close to the original source. New solutions [28, 23] have concentrated on guiding this walking phase while in other solutions [16] the phantom sources are placed in a ring where the messages are mixed with fake traffic.

To hide the presence of events from adversaries with a global hearing range, in [18] all sensors transmit messages at a fixed rate regardless of the existence of real events. This provides perfect privacy but the cost is unacceptable for battery-powered devices. Several authors have focused on reducing the energy waste of this approach. In [29] a filtering scheme is proposed to reduce the amount of fake traffic at various network locations. Also, some statistical approaches [25, 3] have been devised to modify the real and fake transmission frequency without the attacker suspecting.

In general, the presented solutions are based the randomisation of the routes and the injection of fake traffic, which are intended to mislead the adversary or hide the presence of real packets.

2.2. Receiver-Location Privacy

Receiver-location privacy was originally investigated in [8, 9], where various load balancing techniques were designed. A multi-parent routing technique that randomly selects the next hop in the path from all available nodes closer to the base station is proposed. To further complicate traffic analysis, this technique is complemented with random walks in any direction and the injection of fake packets with a given probability distribution.

A similar approach is proposed by Jian et al. [13, 14]. They suggested dividing the neighbours of each node into two groups: closer and further. Later, data packets are sent with higher probability to nodes belonging to the group of closer neighbours. This results in a traffic pattern biased towards the base station, which can be noticed by an attacker after a sufficient number of observations. This problem is reduced by injecting fake packets in the opposite direction with some probability. However, the attacker can still determine whether a packet is fake in some situations and therefore he is able to determine that the base station is in fact, in the other direction.

Other approaches [9, 5] have concentrated on the creation of hotspots, which are areas with high volumes of fake traffic that aim to attract adversaries. The main limitation of these schemes is not only the massive energy waste but also the ability to discard a hotspot once visited. Another scheme based on the injection of fake traffic is [31], where sensor nodes are programmed to transmit the same number of packets regardless of their proximity to the base station so that the traffic rate is homogenised. This strategy provides the best protection but it is also imposes the highest energy requirements because it requires all sensors to constantly inject fake traffic. A different approach is used in [2], which makes the base station mimic the behaviour of ordinary nodes (i.e. forward some of the packets it receives). However, the traffic trend still points to the area of the base station. Additionally, this approach proposes moving the base station to a safer location based on its own measured privacy level but it is not always possible to move it nor easy to determine whether the new position is really safe.

We proposed HISP [21], a packet transmission protocol also based on random route generation and fake packet transmissions that is capable of circumventing some of the problems presented by the previous works. Moreover, HISP-NC extends our preliminary results by incorporating a new mechanism that perturbs the routing tables of the nodes to cope with adversaries performing routing tables inspection. None of the previous solutions have considered this type of attack as a threat to receiver-location privacy.

3. Problem Statement

This section presents the network model as well as the capabilities of the adversary. It also introduces the main assumptions applicable to the rest of this paper.

3.1. Network Model

We consider WSNs used for monitoring purposes. Usually, these types of networks follow an event-driven model, which means that the decision to transmit data to the base station is made by individual sensor nodes immediately after the occurrence of special events in their vicinity. Consequently, this implies a many-to-one communication model where all the information flows from source nodes to a single or just a few base stations.

In this paper we consider networks with a single base station although the robustness of the solution is not affected by the number of base stations. As a matter of fact, having a single base station is the worst case scenario since all the traffic is addressed to a single device resulting in a more abrupt traffic pattern. In a setting with multiple base stations, the amount of traffic is more balanced between all potential recipients. Also, if the goal of the adversary is to bring down the network, he has to destroy each base station and eventually the scenario will be as the one considered here.

Moreover, we assume that the network is comprised of numerous sensor nodes which are deployed over a vast area. This prevents the adversary from both controlling the communications in a large portion of the network and having all sensors within easy reach. On top of that, sensor nodes could be hidden or placed beyond the visual field of the adversary. Sometimes this is not a strong assumption, for example if we consider application scenarios such as under-water or under-ground sensor networks.

We focus on highly-connected sensor networks, where every node is aware of its adjacent neighbouring nodes and the direction towards the sink. This information is achieved by means of a topology discovery protocol, which allows sensor nodes to build their routing tables. The data contained in the routing table might vary depending on the implementation but it must contain information about the distance (e.g., in number of hops) from each neighbour to the base station. In this paper, the routing table is sorted incrementally. More precisely, those neighbours which are closer than the original node to the base station are placed at the top of the table, the neighbours at the same distance are located in the middle, and the neighbours which are one hop further away are placed at the bottom of the table. We denote these groups of nodes as $L^{\mathcal{C}}$, $L^{\mathcal{E}}$ and $L^{\mathcal{F}}$, respectively.

Finally, we assume that each sensor node shares keys with its immediate neighbours and makes use of secure encryption algorithms that prevent an adversary from obtaining any identifiable information from packet payloads. In other words, the encryption mechanism under consideration must be robust to cryptanalysis and also provide indistinguishability between real and fake transmissions. In order to achieve this feature, sensor nodes could add some noise to the payload of their messages before these are encrypted. The noise can be in the form of a secure random sequence [17] or a counter that is incremented for each transmitted packet.

3.2. Adversarial Model

The adversary considered here might take advantage of both traffic analysis and routing tables inspection in order to determine the location of the base station. For the sake of clarity, we assume that the adversary either chooses to perform one of these attacks at a time and thus we describe the capabilities of these attackers separately. Nonetheless, it would be possible for a single adversary to use both sources of information in an attempt to improve his success rate.

3.2.1. Traffic Analysis Attacks

Traffic analysis attacks consists of extracting or inferring information based on the mere observation of the traffic traversing the network. Consequently, adversaries performing this type of attack can be categorised mainly based on their eavesdropping capabilities and the mechanisms they use to extract the information from their observations.

First, we consider the eavesdropping capabilities of the adversary. In particular, we concentrate on both the hearing range and the ability to retrieve packet header information. With respect to the hearing range, adversaries might range from those capable of observing the transmissions of a single node to those powerful enough to monitor all the communications in the network. On the other hand, we distinguish between adversaries who, by observing a message, are capable of recognising the addressee of the next hop and those unable to retrieve this information. This information is contained in the header of the packets but it might be protected by means of some sort of pseudonyms mechanism [6]. Next, we provide a formal definition of the adversarial model:

Definition 1 ($\mathcal{ADV}$)**.** *Let $X = \{x_1, x_2, \cdots, x_m\}$ be the set of sensor nodes comprising the network and let x_i be an ordinary sensor node in the proximity of the adversary. We define the following adversaries:*

- $\mathcal{ADV}_n$ *chooses first a node x_i, and then observes the transmissions of node x_i and all its neighbours within distance n. In the next round he may choose a different node $x_{i'}$. The choice of the next $x_{i'}$ depends on the movement strategy, see for instance time-correlation and rate monitoring, below.*
- $\mathcal{ADV}_n^a$ *is similar to the previous one: he observes the transmissions of node x_i and all its neighbours within distance n, but this observation includes also the addressees of all those transmissions.*

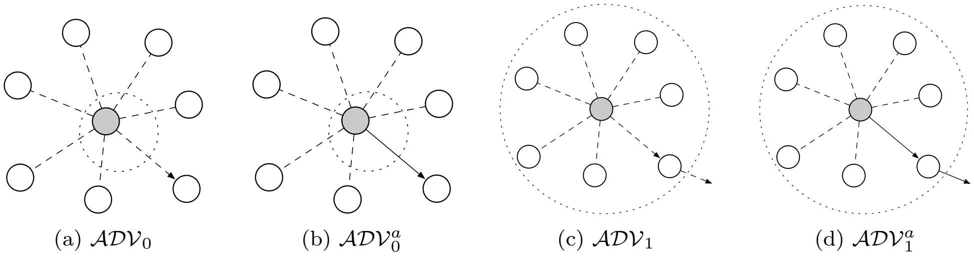

(a) $\mathcal{ADV}_0$ (b) $\mathcal{ADV}_0^a$ (c) $\mathcal{ADV}_1$ (d) $\mathcal{ADV}_1^a$

Figure 2: Adversarial Model Examples

We could define other types of attackers that are unable to monitor all the neighbours within a certain distance but only a partial set of them. However, these types of attackers and their analysis will be left for future work.

The attacker model considered here has a *local* hearing range, similar to an ordinary sensor node[1] (i.e., $\mathcal{ADV}_1$), which is the typical hearing range considered in the literature. These adversaries are capable of monitoring any packets transmitted by nodes at distances no larger than 1, as those depicted in Figure 2. In this figure, the central node, x_i, broadcasts a message that is received by all its immediate neighbours. Transmissions are depicted by means of lines and arrows. An arrow represents that the packet is addressed to that particular node while dashed lines represent that these nodes are passive observers. When the arrow is dashed we mean that the node identifier cannot be retrieved by the attacker while the ordinary arrow represents that the identifier is accessible. Finally, the dotted circles represent the hearing range of the adversary.

Moreover, the adversary is *mobile* and decides in which direction to move based on his observations and the particular features of the communication model. Also, we assume that the attacker knows how the protection mechanism works or he will eventually understand it (i.e., we adopt Shannon's maxim [24]). When the adversary reaches the next node he continues to analyse the traffic in his vicinity in an attempt to reduce the distance to the base station. Deciding which node to visit next is based on the information gathered from the two attack strategies proposed in the literature: the

[1] The hearing range of current sensor nodes operating outdoors is around 100 meters for low power configurations [12]. However, these values might be altered by many factors such as the signal frequency or the presence of obstacles.

time-correlation and the rate-monitoring attack.

In a *time-correlation* attack, the adversary observes the transmission times of a node x_i and its neighbours. Based on the assumption that a node forwards a received packet shortly after receiving it, the adversary is able to deduce the direction to the sink and move accordingly. In a *rate-monitoring* attack, the adversary moves in the direction of the nodes transmitting a higher number of packets. This attack is based on the fact that nodes in the vicinity of the base station must transmit not only their own data packets but also the traffic from remote sources. This strategy is less efficient than the previous one because it means the adversary has to capture a sufficient number of packets before moving. Additionally, this attack is not effective when there are very few data sources or the adversary is not close to the sink.

3.2.2. Routing Tables Inspection

Node capture is a form of physical attack which is favoured by the unattended nature of sensor networks. Sensor nodes are usually deployed in open and hostile environments and thus they are within reach of adversaries which might try to tamper with them. Physical attacks may come in various guises [4] that range from node destruction to node reprogramming as well as node replacement or the extraction of data contents from the memories in the node.

Here, we concentrate on adversaries who capture sensor nodes with the sole purpose of retrieving information that might be useful for reaching the base station. The goal of the adversary is not to destroy the nodes or modify their software to interfere with the communications or the normal operation of the network. This allows the attacker to remain undetected to potential intrusion detection systems and therefore continue tracking down its target for a longer period of time. Note that this is also the case for adversaries performing traffic-analysis attacks. In general, we say that the adversarial model considered in this paper is *passive*.

After capturing a node, the adversary may have access to the data contained in the node. In particular, the most valuable piece of information for an adversary willing to reach the base station is the routing table. A node's routing table indicates the distance from each of its neighbours to the base station (see Figure 3), which is used to select the most suitable routing paths. Consequently, an adversary retrieving the routing tables of several nodes may acquire a very good clue as to the distance and direction towards the base station.

In this respect, the node capture strategy is not clearly defined in the literature because, as far as we are concerned, this is the first receiver-location privacy solution to consider this threat. Nonetheless, several papers have dealt with the modelling and mitigation of node capture attacks in WSNs (e.g., [7, 26]) particularly in the protection of secure communication channels for random key distribution systems. Some authors consider that adversaries pick nodes in the field at random while others assume that the adversary chooses to compromise (all or some) nodes within a particular region. In this work we consider that the adversary is more successful if he captures nodes from nearby, rather than randomly. Also some features, such as the time it takes to compromise a single node, are considered by other authors but we will not take them too much into consideration here. Instead, we will assume that the adversary is not capable of inspecting more than a given number of routing tables during a single data transmission phase.

Once the adversary has captured a node and retrieved its routing table he can make a decision on his next move. Provided that the routing tables are correct, the adversary is certain that the first neighbour in the table is closer than the current node to the base station. Thus, the adversary is more likely to reach the base station if he moves towards the first neighbour in the routing table for each compromised node. Moreover, after only a few captures the adversary obtains a good idea of the direction towards the base station. More details about the operation of the adversary are provided later in Section 7.

4. Data Transmission Scheme

This section provides a detailed description of the HISP-NC data transmission protocol. We present an overview of its main features as well as some fundamental properties that must hold so as to ensure a robust privacy-preserving transmission protocol and the arrival of packets to the sink. Also, the neighbour discovery process is described since it is crucial for the subsequent data transmission stage.

4.1. Overview

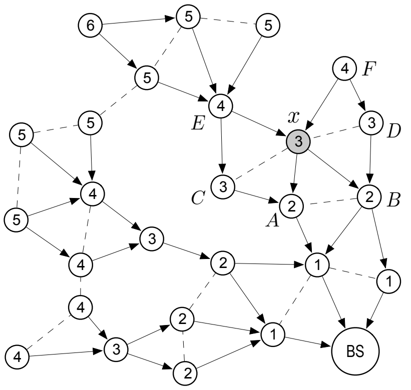
The transmission protocol used by HISP-NC is basically a biased random walk scheme reinforced with the injection of controlled amounts of fake traffic. Whenever a node has something to transmit, it sends two packets to different random nodes. This probabilistic process is repeated for each transmission and it is devised to ensure that messages flow in any direction; evenly distributing the traffic among all neighbours. One of these packets is more likely sent to a node closer to the base station while the other packet is addressed to a neighbour at the same distance or further away with high probability. Consequently, one of the packets can carry real data and the other one can be used as a mechanism to hide the data flow. In this way, the protocol prevents the adversary from successfully determining the direction to the sink by observing the packets transmitted in his vicinity while the delivery delay is not significant.

This process is guided by a computationally inexpensive approach that determines the recipients of messages. A node selects two neighbours by picking uniformly at random, a combination resulting from all the combinations of two elements without repetitions from its routing table. Since the routing tables are arranged in such a way that the nodes closer to the base station are at top of the table, the resulting combinations are more likely to have one of these nodes in the first position of the duple. Therefore, real packets are sent to the first node in the combination and fake packets are sent to the second. As each node appears in the same number of combinations, the traffic is evenly distributed. Moreover, nodes receiving fake traffic must also send two messages, both of which are fake, to prevent the protocol from leaking information. Also, a time-to-live parameter is introduced to control the durability of fake traffic in the network.

4.2. Neighbour Discovery Process

Shortly after the deployment of the network, a network discovery protocol is launched to allow sensor nodes to route data packets. This process is initiated by the base station, which broadcasts a message containing a hop count initially set to zero. On reception, each node stores the minimum hop count received from its neighbours and forwards the message after increasing the hop count by one. In this way, each node builds a routing table that contains its neighbours at distance $n - 1$, n, and $n + 1$, where n is the number of hops from the node to the base station. The result of this process is depicted in Figure 3. In this figure we represent a particular network configuration and the routing table[2] of node x, which is three hops away

[2] It is not necessary to keep the distance or group values within the table. The arrangement (ordering) is sufficient for our protocol to work.



neighs(x)	distance	group
A	$n-1$	$L^{\mathcal{C}}$
B	$n-1$	
C	n	$L^{\mathcal{E}}$
D	n	
E	$n+1$	$L^{\mathcal{F}}$
F	$n+1$	

Figure 3: Routing table of shaded node x

from the data sink. This node may use nodes A or B, which are one hop closer to the base station, as data relays.

The neighbour discovery process is essential to the rest of our protocol. The reason is that the number and distribution of neighbours affects to both the level of protection and the delivery time as we will show in the following sections.

4.3. Data Transmission Properties

The protocol we are aiming for uses both real and fake messages. The source node, as well as any node that receives a real message, sends a real and a fake message, which should be indistinguishable to an adversary but not to the addressees. Property 2 aims to balance the amount of traffic being delivered from a node to its neighbours. By doing this, a local adversary cannot make a decision on which direction to follow based on the number of packets forwarded to neighbouring nodes. While the paths of fake messages are relatively short (this is a parameter of the solution), the path of real messages is intended to converge on the sink. This is established by Property 1: real messages must be transmitted to nodes closer to the base station with a high probability. These two properties together ensure that both real packets reach the base station and also that the flow of real messages is hidden by fake messages since they are indistinguishable. An additional technical property ensures that the transmission of each pair of messages is sent to two different nodes.

Property 1 (Convergence)**.** *Let x be an arbitrary sensor node and BS be the base station. Also, let $neigh(n)$ be the set of immediate neighbours of a particular node n. Then we say that a path is convergent if x chooses the next node $x' \in neigh(x)$ such that:*

$$E(dist(x', BS)) < E(dist(x, BS))$$

where E is the mathematical expectation and $dist$ is the distance between two particular nodes.

Property 2 (Homogeneity)**.** *Let x be an arbitrary sensor node and $neigh(n)$ be the set of immediate neighbours of a particular node n. We say that the transmissions of a node x hold the homogeneity property if:*

$$\forall y, z \in neigh(x) \quad Frec_m(x, y) \simeq Frec_m(x, z)$$

where $Frec_m(x, y)$ represents the number of messages (real and *fake) transmitted by node x to node y.*

Property 3 (Exclusion)**.** *Let m and m' be a pair of messages and t be a particular transmission time. Let $send(m, x, y, t)$ denote that x sends to y the message m at time t. The exclusion property states that:*

$$\forall m, m', x, y, t \quad send(m, x, y, t) \wedge m \neq m' \Rightarrow \neg send(m', x, y, t)$$

The fulfilment of all these properties guarantee the usability of the system and privacy of the base station. Next, a data transmission protocol that is consistent with these properties is presented.

4.4. Transmission Protocol

The HISP-NC data transmission protocol introduces insignificant computational and memory overhead because it is based on straightforward operations. More precisely, it only requires a simple sorting operation and a pseudo-random number generator [15].

Since we need to send two messages, the combinations of two elements without repetition from all neighbours in the routing table is an elegant and lightweight mechanism for the selection of neighbours, which is conforms to the provisions of Property 3. Moreover, if the routing table is sorted incrementally in terms of the distance of its neighbours to the base station (i.e.,

Algorithm 1 Transmission strategy

```
Input: $packet \leftarrow receive()$
Input: $combs \leftarrow combinations(\{L^{\mathcal{C}}, L^{\mathcal{E}}, L^{\mathcal{F}}\}, 2)$
Input: $FAKE\_TTL$
1: $\{n1, n2\} \leftarrow select\_random(combs)$
2: if $isreal(packet)$ then
3:     $send\_random(n1, packet, n2, fake(FAKE\_TTL))$
4: else
5:     $TTL \leftarrow get\_time\_to\_live(packet) - 1$
6:     if $TTL > 0$ then
7:         $send\_random(n1, fake(TTL), n2, fake(TTL))$
8:     end if
9: end if
```

$[L^{\mathcal{C}}, L^{\mathcal{E}}, L^{\mathcal{F}}]$) we can ensure that most of the resulting combinations have a closer or equally distant neighbour in the first position. Therefore, Property 1 is satisfied if the real packet is always transmitted to the first neighbour. Also Property 2 holds provided that we pick a combination uniformly at random from the set of all possible combinations.

In Algorithm 1 we describe the behaviour of a node upon the reception of a packet. The algorithm uses as input the received packet, a data structure that contains the combinations of two neighbours from the routing table, and a network parameter that controls the durability of fake packets in the network. Initially, the algorithm decides the random pair of neighbours to whom packets will be addressed (line 1). Subsequently, if the received packet is real, then it is be forwarded to $n1$ while $n2$ receives a fake packet whose time-to-live is set to $FAKE\_TTL$ (line 3). This parameter is dependent on the hearing range of the adversary and provides a trade-off between energy consumption and privacy protection. Also, note that the packets are sent in random order to prevent the adversary from trivially learning which is the real message. The described behaviour is identical in the case the node, rather than being an intermediary, is a source node which signals the occurrence of a special event in the field.

To the contrary, if the received packet is fake, the node first obtains the time-to-live (TTL) from the packet and decrements its value by one (line 5). In case the new TTL is greater than zero, the node sends two fake messages with the current TTL value (line 7). Since we consider adversaries with a hearing range similar to an ordinary sensor node (i.e., the family $\mathcal{ADV}_1$), fake messages might be forwarded only once but still exceed the reach of the adversary. This mechanism prevents fake messages from flooding the network and at the same time impedes adversaries from obtaining information from non-forwarded fake packets.

5. Routing Table Perturbation Scheme

This section describes the routing table perturbation scheme implemented by HISP-NC. First, we overview the need for and main features of the proposed solution. Then we present some naive solutions to the routing inspection problem and establish some perturbation requirements. Finally, we describe the devised perturbation algorithm.

5.1. Overview

Routing tables are a fundamental component of almost any data transmission protocol. They contain relevant information regarding the location or distance to the data sink. The HISP-NC data transmission protocol relies on the order of the table to determine suitable combinations of neighbours that satisfy the usability and privacy of the system. However, the traffic normalisation efforts could be rendered completely useless if an adversary can inspect the routing tables as he would be able to determine which nodes are closer to the base station regardless of the use of traffic analysis techniques.

The routing table perturbation scheme complements the data transmission scheme by introducing some modifications to the routing tables of the sensor nodes. The modifications consist of a re-arrangement of the table in such a way that neighbours closer to the base station are not necessarily at the top of the table, neighbours at the same distance are not compulsorily in the middle, and likewise neighbours further away are not always at the bottom. In this way if an adversary captures the routing table of a node he cannot be certain of which neighbours in the table are closer to the base station.

The devised perturbation algorithm is modelled as an optimisation problem and it is inspired on evolutionary strategies to find a solution. The algorithm is guided by a simple parameter that controls the degree of perturbation applied on the routing tables. This parameter balances between the efficiency of data transmission protocol and the resilience to routing table inspection.

5.2. Basic Countermeasures

The original distribution of the routing tables used by the HISP-NC transmission protocol is such that neighbours closer to the base station are placed before neighbours at the same distance, and these in turn are placed before farther neighbours (recall Figure 3). This particular arrangement of the table is important for the generation of combinations of two neighbours where the first element is highly likely to be in the set of closer nodes (i.e., $L^{\mathcal{C}}$), which allows the distance to the base station to be reduced. However, if an attacker retrieves the routing table of a node he might use this information to determine which neighbours are closer, move to any of these nodes, and repeat the process. After very few repetitions the attacker has a very good estimation of the direction towards the base station. To prevent node routing table inspection from being a substantial threat to receiver-location privacy, it is necessary to introduce some uncertainty into the routing tables

Since the routing tables of the nodes may change after each topology discovery protocol, the perturbation must be performed on all sensor nodes. Otherwise, if the decision of modifying the routing tables was determined by a particular probability distribution, the adversary could compromise a node and wait until the next discovery phase to check whether its routing table has changed. If so, the adversary only needs to wait for a sufficient number of updates until he discovers the pattern. In fact, the number of updates does not have to be necessarily high since observing the same routing table after a few discovery phases, indicates with a high probability that the original table is this one. To further increase the chances of correctly learning the real routing table, the adversary only needs to make more observations. In the long term, the original routing table stands out from the modified versions.

Similarly, making the nodes store fake routing tables does not provide extra protection to the real table. There are two main reasons why this is not an effective protection mechanism. On the one hand, the sensor node must also store a variable or pointer to the actual routing table and this information would be available to an adversary as well. On the other hand, even if it is not easy to determine which is the real routing table by analysing the memory of node (i.e., because it is obfuscated in someway), the attacker can eventually identify which table is in use. For example, this information can be retrieved by comparing the sets of routing tables generated by a node after different discovery phases. Those elements not present in the intersection of the sets of tables from different phases can be discarded if we consider sensor nodes to be honest in the sense that the real routing table is always contained in the node.

5.3. Perturbation Requirements

The routing table of each node must be perturbed to prevent an attacker from easily gaining information about the location of the base station after inspecting them. The routing table resulting from the perturbation algorithm must (i) provide a sufficient level of uncertainty in the adversary and still (ii) be usable to enable the arrival of data packets at the base station.

Next we provide a formal definition of a routing table that will be later used to prove some desirable properties of the devised perturbation algorithm.

Definition 2 (Routing table)**.** *Let* $L^* = L^{\mathcal{C}} \cup L^{\mathcal{E}} \cup L^{\mathcal{F}}$ *be the list of all the neighbours of a node* n*, where* $L^{\mathcal{C}} = \{c_1, c_2, c_3, \ldots\}$ *are neighbours of level* $n-1$*,* $L^{\mathcal{E}} = \{e_1, e_2, e_3, \ldots\}$ *are neighbours of level* n*, and* $L^{\mathcal{F}} = \{f_1, f_2, f_3, \ldots\}$ *are neighbours of level* $n+1$*.*

A routing table is a bijection $r : \{N-1, \ldots, 0\} \rightarrow L^*$*, being* N *the total number of neighbours.*

In other words, a routing table is simply an ordering of all the neighbours of a specific node. Similarly, we can define $pos : L^* \rightarrow \{N-1, \ldots, 0\}$ as the inverse of r, such that, given a specific neighbour it returns the position of this node in the table. An example is depicted in Table 1, where $pos(c_1) = N-1$, $pos(f_3) = N-2$, and so forth.

Position		Node
$N-1$	→	c_1
$N-2$	→	f_3
$N-3$	→	c_2
…		…
1	→	e_6
0	→	f_5

Table 1: A Specific Arrangement of a Routing Table

Having gained the previous definitions we are in a position to determine in which circumstances a routing table enables the eventual delivery of data packets to the base station. When these conditions are met we say that the routing table is correctly biased.

Theorem 1. *A routing table is correctly biased iff* $\sum\limits_{n \in L^{\mathcal{C}}} pos(n) > \sum\limits_{n \in L^{\mathcal{F}}} pos(n)$

More simply, a routing table is correctly biased if and only if the probability of choosing an element from $L^{\mathcal{C}}$ as the recipient of data packets is higher than the probability of choosing an element from $L^{\mathcal{F}}$.

Proof. Assume that we pick a random combination of neighbours (n_1, n_2), where $pos(n_1) > pos(n_2)$ as defined by our data transmission protocol. Given a subset $L \subseteq L^*$ we want to know what the probability is that the first node, n_1, is in L. This probability is given by the following expression:

$$\mathbb{P}(n_1 \in L) = \frac{1}{C} \sum_{n \in L} pos(n) \tag{1}$$

where $C = N * (N-1)/2$ is the total number of combinations of two elements without repetition of L^*. Also note that $C = 1 + 2 + \ldots + (N-1) = \sum\limits_{n \in L^*} pos(n)$.

It is possible to write all possible combinations without repetitions of two nodes as a list of pairs, lexicographically ordered, from the routing table:

$$\begin{array}{rrrrr}
(r(N-1), r(N-2)), & (r(N-1), r(N-3)), & (r(N-1), r(N-4)), & \ldots, & (r(N-1), r(0)) \\
 & (r(N-2), r(N-3)), & (r(N-2), r(N-4)), & \ldots, & (r(N-2), r(0)) \\
 & & (r(N-3), r(N-4)), & \ldots, & (r(N-3), r(0)) \\
 & & & & \ldots \\
 & & & & (r(1), r(0))
\end{array}$$

Since the node $r(N-1)$ appears in the first position of $N-1$ pairs, the node $r(N-2)$ in $N-2$ pairs, and so on, they are exactly $(N-1) + (N-2) + (N-3) + \ldots + 1$ pairs in the list, which is $N * (N-1)/2 = C$.

Now, choosing a random pair (n_1, n_2) such that $pos(n_1) > pos(n_2)$ is equivalent to choosing any pair from the previous list. Thus, the probability that a certain node n_1 is chosen as the first entry is simply the number of elements in the routing table r whose position is below n_1, divided by the total number of pairs. This is precisely $pos(n_1)/C$ and Equation 1 follows directly.

□

The perturbation degree or bias of a routing table, $bias(r)$, is an important parameter to quantify because it determines both the speed of convergence of data packets to the base station and the uncertainty level of the attacker. We define the bias of a routing table r, $bias(r) \in [-1, 1]$, as the probability of sending data packets in the direction of or in the opposite direction to the base station. This parameter compares the level or distance of the current node, $level(n_0)$, with the expected value of the level of the next node in the transmission path, i.e., $E(level(n_1))$. The closer the bias is to 1 the greater the probability is that data packets are sent to nodes in $L^{\mathcal{C}}$ (i.e., the distance decreases). Likewise, a bias value close to -1 implies that it is highly likely that the first element of the resulting combination belongs to $L^{\mathcal{F}}$.

The bias of a routing table can be calculated as the weighted difference between number of combinations resulting from the neighbours in $L^{\mathcal{C}}$ and the number of combinations resulting from neighbours in $L^{\mathcal{F}}$. Formally:

$$bias(r) = \frac{1}{C}(\sum_{n \in L^{\mathcal{C}}} pos(n) - \sum_{n \in L^{\mathcal{F}}} pos(n)) \tag{2}$$

Proof. By definition, we have that the bias of a routing table is:

$$bias(r) := level(n_0) - E(level(n_1))$$

The level of the next node n_1 is the same level as n_0, or this value decremented or incremented by 1. This is determined by the list of neighbours to which the node belongs, $L^{\mathcal{E}}$, $L^{\mathcal{C}}$, or $L^{\mathcal{F}}$, respectively. Thus,

$$\begin{aligned} E(level(n_1)) &= (level(n_0) - 1) * \mathbb{P}(n_1 \in L^{\mathcal{C}}) + \\ &\quad (level(n_0) + 1) * \mathbb{P}(n_1 \in L^{\mathcal{F}}) + \\ &\quad level(n_0) * \mathbb{P}(n_1 \in L^{\mathcal{E}}) \\ &= level(n_0) - [\mathbb{P}(n_1 \in L^{\mathcal{C}}) - \mathbb{P}(n_1 \in L^{\mathcal{F}})] \end{aligned}$$

and now the result follows directly from Equation 1.

□

As previously defined, the bias is a value in the $[-1, 1]$ interval, but not all values are eligible because the bias is dependent on the number of elements in $L^{\mathcal{C}}$ and $L^{\mathcal{F}}$. For example, $bias(r) = -1$ if and only if $L^* \equiv L^{\mathcal{F}}$, since $L^{\mathcal{C}} = \emptyset$ and $\sum_{n \in L^{\mathcal{F}}} pos(n) = C$.

Let us first calculate the upper bound of the bias. The maximum value, $bias_M(r)$, is reached when the elements in $L^{\mathcal{C}}$ are placed at the top of the routing table, the elements in $L^{\mathcal{F}}$ are placed at the bottom, and the elements in $L^{\mathcal{E}}$ are in between. Consequently, Equation 2 can be written in the following form:

$$bias_M(r) = \frac{1}{C}(\sum_{i=1}^{c}(N-i) - \sum_{i=1}^{f}(i-1)) \tag{3}$$

where c, f, and N are the number of elements in $L^{\mathcal{C}}$, $L^{\mathcal{F}}$, and L^*, respectively.

Similarly, the minimum value, $bias_m(r)$, is reached when $L^{\mathcal{C}}$ is at the bottom, $L^{\mathcal{F}}$ at the top, and $L^{\mathcal{E}}$ in the middle. Then, we can define it as:

$$bias_m(r) = \frac{1}{C}(\sum_{i=1}^{c}(i-1) - \sum_{i=1}^{f}(N-i)) \tag{4}$$

After mathematical transformations we have that the bias of a particular routing table r is bounded by the following equation:

$$\frac{c(c-1) - 2fN + f(f+1)}{N(N-1)} \leq bias(r) \leq \frac{2cN - c(c+1) - f(f-1)}{N(N-1)} \tag{5}$$

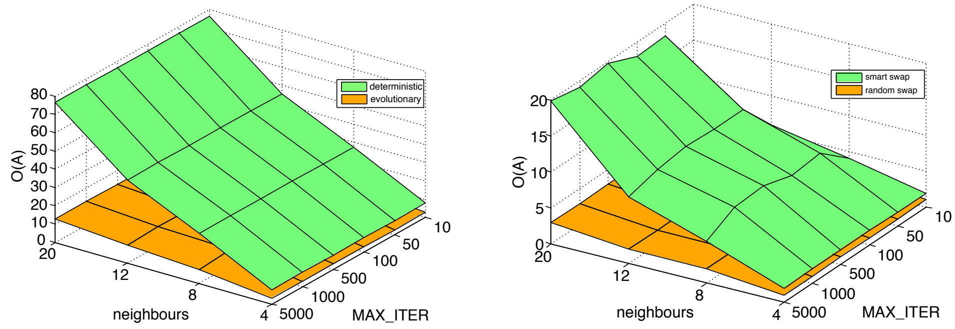
5.4. Perturbation Algorithm

The perturbation algorithm in HISP-NC receives a routing table and a desired value and outputs an ordering of the table that satisfies, to some degree, the input bias value. This algorithm must be implemented by all nodes in the network and given the hardware limitations of the nodes, the complexity of the algorithm (i.e., completion time and memory requirements) must be minimised.

A deterministic perturbation algorithm that explores the entire search space of solutions has a complexity of $\mathcal{O}(\mathcal{A})$:

$$\mathcal{O}(\mathcal{A}) = \frac{N!}{c!\ e!\ f!}$$

where N is the total number of elements in the routing table, and c, e, and f is the cardinality of the groups $L^{\mathcal{C}}, L^{\mathcal{E}}$, and $L^{\mathcal{F}}$, respectively. This sort of algorithm always finds the best solution but the cost is determined by the total number of elements in $L^{\mathcal{C}}, L^{\mathcal{E}}$, and $L^{\mathcal{F}}$. Consequently, such a



(a) Deterministic vs. Evolutionary

(b) Random swap vs. Smart swap

Figure 4: Complexity of perturbation algorithms

deterministic algorithm might be viable for configurations where the total number of elements in the table is low or when the neighbours in the lists are unevenly distributed, i.e., most of the elements belong to a single list.

Alternatively, this problem can be modelled as an optimisation algorithm where the objective function depends on the desired bias of the table and the positions of the nodes comprising it. More precisely, our algorithm is inspired by evolutionary strategies [11] where simple mutations are applied to the routing table in order to minimise the distance to the desired bias. In Figure 4a we compare the order of complexity of a deterministic version of the algorithm versus its evolutionary counterpart. We use the decimal logarithmic scale because the number of iterations required for the deterministic algorithm (upper plane) to reach the solution is significantly larger than for the evolutionary algorithm (lower plane). On the y-axis we depict the maximum number of iterations that the evolutionary algorithm[3] is allowed to run since it might never find the best solution. Actually, the results presented for the evolutionary algorithm do not represent the last iteration of the algorithm but rather the last iteration when the value of the objective function was reduced, i.e., the iteration when the algorithm obtained the pseudo-optimal solution.

The perturbation algorithm (see Algorithm 2) is triggered immediately after the topology discovery phase. It receives as input the lists of neighbours from levels $n-1$, n, and $n+1$ as well as the desired bias for the routing table

[3] The deterministic algorithm is not affected by this variable.

Algorithm 2 Perturbation Algorithm

```
Input: $br \leftarrow \{L^{\mathcal{C}}, L^{\mathcal{E}}, L^{\mathcal{F}}\}$
Input: $bias$, $MAX\_ITER$
 1: $E \leftarrow energy(bias, br)$
 2: $i \leftarrow 0$
 3: while $(i < MAX\_ITER) \wedge (E \neq 0)$ do
 4:     $br' \leftarrow swap(br)$
 5:     $E' \leftarrow energy(bias, br')$
 6:     if $(E' < E)$ then
 7:         $br \leftarrow br'$
 8:         $E \leftarrow E'$
 9:     end if
10:     $i \leftarrow i + 1$
11: end while
12: return $br$
```

and the maximum number of iterations to run. Firstly, the algorithm calculates the distance to the objective by means of the *energy* function (line 1). This function is basically defined as the distance between the desired bias and the bias of the current ordering of the table. The operations performed from line 3 to line 11 are intended to reduce the aforementioned distance. To that end, a mutation is performed over the current routing table (line 4) and then its energy is calculated. The mutation consists of swapping two elements of the table using a particular strategy. In the case that the mutation reduces the value of the objective function, then the previous routing table is discarded. The process is repeated for MAX_ITER iterations or until the desired bias is reached. Finally, the algorithm returns the perturbed routing table but before starting the data communication phase, the node must securely erase any data used by the algorithm.

Figure 4b depicts the performance of our algorithm with two different *swap* functions, which gives a good idea of the average number of iterations our algorithm needs to find a pseudo-optimal solution. More precisely, the upper plane represents the median number of iterations when using a function that swaps two random elements of the table. In contrast, the lower plane represents the mean number of iterations when the mutation is more intelligently done and consists of swapping the two elements that achieve the largest decrease on the value of energy function. Clearly, as shown in the figure, the smart swapping converges faster on the solution than the random swapping, especially as the number of neighbours increases, but it requires more processing power.

Finally, note that this algorithm might not reach the optimal solution but it converges to it. Either it is infeasible to achieve the expected solution for the given lists of neighbours (see Equation 5), or the number of iterations of the algorithm was insufficient for the swapping function to allow the convergence. Also, given the non-deterministic nature of the solution, it may be that the result differs for two runs of the algorithm with the same input parameters. This provides an extra means of protection from reversing attacks.

6. Protocol Analysis

This section presents a detailed analysis on the potential limitations that might hinder the successful operation of the HISP-NC protocol. The numerical analysis is supported by a number of experiments conducted in our own discrete-event simulation environment developed in Matlab [1]. First, we explore the impact of the network topology and the expected number of hops for real messages to reach the base station prior to and after the perturbation of the routing tables. Finally, we analyse the overhead introduced by our solution in terms of fake packet transmissions.

6.1. Network Topology

The distribution of real and fake messages is clearly impacted by the number of the neighbours in each of the groups of the routing table of the nodes. As stated in Section 5.3, the arrangement of the table and the size of each of the groups of neighbours determine the bias of the table. In other words, Property 1 could be unsatisfied if the number of neighbours in $L^{\mathcal{C}}$ is significantly lower than the number of neighbours in $L^{\mathcal{F}}$. This problem is dependent on the topology of the network and the hearing range of the nodes.

To have a clearer picture as to what extent this poses a real limitation to our data transmission protocol, we provide a numerical analysis on the number of elements in $L^{\mathcal{F}}$ that any sensor node can withstand without sacrificing the usability and privacy of the system. The present analysis considers the unperturbed version of the routing table, where the elements are arranged according to their distances to the base station.

Let N be the total number of neighbours of an arbitrary node such that $N = c + e + f$, where c, e, and f are the number of neighbours in $L^{\mathcal{C}}$, $L^{\mathcal{E}}$, and $L^{\mathcal{F}}$, respectively. The theorem below gives a sufficient condition on c, f and N to ensure the desired property of data convergence.

Theorem 2. *Real messages follow a biased random walk converging to the base station if $f < \sqrt{2c(N-c)}$ for any sensor node in the route.*

Proof. We want to show that if $f < \sqrt{2c(N-c)}$ then $\mathbb{P}(n_1 \in L^{\mathcal{C}}) > \mathbb{P}(n_1 \in L^{\mathcal{F}})$, which represent the probabilities of sending a data message to a node in $L^{\mathcal{C}}$ and $L^{\mathcal{F}}$, respectively.

The number of combinations of two neighbours where at least the first element belongs to $L^{\mathcal{F}}$ is:

$$\binom{f}{2} = \frac{f(f-1)}{2}$$

while the number of combinations of two neighbours where the first element of the duple is a node in $L^{\mathcal{C}}$ is:

$$\binom{c}{2} + c(e+f)$$

Consequently, the probability of selecting a neighbour in $L^{\mathcal{C}}$ is higher than the probability of selecting a neighbour $L^{\mathcal{F}}$ iff the number of combinations with a closer neighbour in the first position of the duple is larger than those with the first element being a further neighbour. Formally:

$$\mathbb{P}(n_1 \in L^{\mathcal{C}}) > \mathbb{P}(n_1 \in L^{\mathcal{F}}) \Leftrightarrow c(c-1) + 2c(e+f) > f(f-1)$$

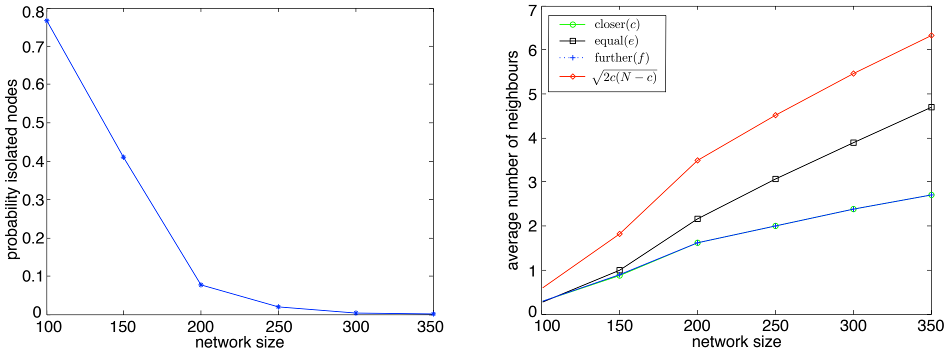
In order to simplify the analysis we make some generalisations which are less restrictive but still provide a sufficient condition for the proof.

$$2c(e+f) > f^2 \Rightarrow c(c-1) + 2c(e+f) > f(f-1)$$

Provided that $c + e + f = N$, the previous equation can be expressed as:

$$f < \sqrt{2c(N-c)} \tag{6}$$

Therefore, we might say that if Equation 6 is satisfied, then the following implication holds:



(a) Probability of isolated nodes

(b) Neighbours restriction

Figure 5: Node connectivity in randomly deployed networks

$$f < \sqrt{2c(N-c)} \Rightarrow \mathbb{P}(n_1 \in L^{\mathcal{C}}) > \mathbb{P}(n_1 \in L^{\mathcal{F}})$$

□

Intuitively, the imposed restriction can be satisfied in manually deployed networks deployed following a particular topology (e.g., grid or mesh). Yet we deem it necessary to validate the feasibility of our restriction in randomly deployed networks by means of experimental simulations. In particular, Figure 5 depicts the average results over 50 repetitions of our network discovery protocol for various network sizes. We considered the following network parameters: (i) a square field area of side 1, (ii) the transmission radius of the nodes is set to 0.1, and (iii) networks ranging in size from 100 to 700 randomly deployed nodes. In Figure 5a we show that the probability of isolated nodes drops significantly when the network size is over 200 nodes. Moreover, Figure 5b presents the average number of neighbours closer, equal and farther for any node in the network. In this figure we also show that the restriction imposed by Equation (6) on the maximum number of further neighbours is satisfied at all times.

Note that the results shown in Figure 5b are average values and there might be some nodes not satisfying the restriction. However, this would only pose some additional delay unless there are network regions with a high concentration of nodes unable to fulfil the imposed condition. This issue might cause network packets to continuously move back and forth impeding their progress towards the base station. This is not the case when the node density is sufficient. However, this is a problem that does not only affect our solution.

In general, we can state that when the density of a randomly deployed network is over 350 nodes per square kilometre there is a high probability of full connectivity; considering transmission ranges of 100 meters. Also, the restriction on the number of neighbours is satisfied for such density.

6.2. Message Delivery Time

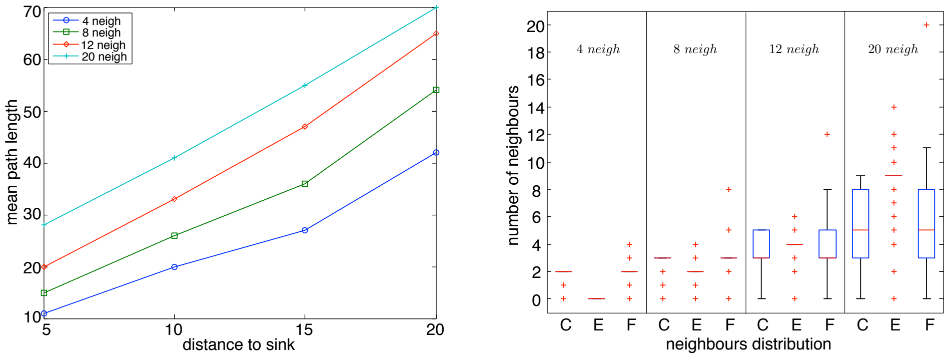
The probabilistic nature of our protocol introduces some uncertainty on the delivery of messages to the sink. This issue has some implications both on the reaction time of the network and the energy consumption of the nodes. Therefore, we provide some insights into the expected number of hops to reach the base station for a packet originated n hops away.

Let x_n be the expected number of hops for a packet originated at distance n. The proposed transmission protocol can be modelled by the following recurrence equation:

$$x_n = 1 + px_{n-1} + qx_n + rx_{n+1} \tag{7}$$

This equation represents a biased random walk where the packet will be forwarded to a neighbour after increasing the number of hops by one. At each hop, we have a probability p of delivering the packet to a node closer to the base station, a probability q of staying at the same distance, and a probability r of moving in the opposite direction. Therefore, the average speed towards the base station is $p - r$.

In general, this result is true for constant values of p and r but this is not always the case in sensor networks. The reason is that not all sensor nodes present the same distribution of neighbours. This depends on the hearing range of the nodes, the network topology and their location in the field. In Figure 6 we present the performance of our protocol for WSNs deployed in a grid with equal transmission power for all nodes. We examine various configurations which are obtained by increasing the transmission power of the nodes and this in turn changes the connectivity of the network. Each of these configurations present, on average, 4, 8, 12 or 20 neighbours per node. Also, for each configuration we place the source at various distances from the base station: 5, 10, 15 and 20 hops. Several source nodes are selected for each distance and every single source node generates 500 data packets to be received by the base station.



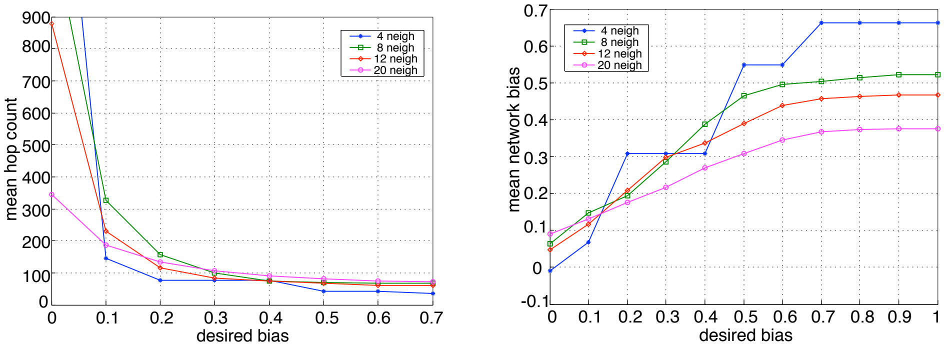
(a) Expected number of hops

(b) Distribution of neighbours

Figure 6: Protocol performace for various network configurations

The results show that the expected number of hops increases with the distance to the sink as well as with the connectivity of the nodes. As the number of neighbours available to a node increases, the more difficult it is for the adversary to make a decision on which of the recipients is actually closer to the base station. However, a significant increase in the number of neighbours also has implications on the delivery time because as the transmission range grows, more nodes are included in the group of equally distant neighbours (i.e., $L^{\mathcal{E}}$) of the node. This issue is shown in Figure 6b, where we provide a box-plot representation of the number of neighbours closer (C), equal (E), and further (F) for the simulated network configurations. For example, C_4 indicates the number of closer neighbours in the $4neigh$ network configuration.

Additionally, note from Figure 6a that, for all the configurations, the average speed of the packets decreases when they are close to the sink. Consider, for example, the $4neigh$ configuration. When the distance to the sink is 5, the expected delivery time is 11, while a packet at distance 20 will be delivered after 42 hops. This means that the time difference from distance 20 to 5 is 31 and thus, the average speed is $15/31 = 0.484$. However, in the proximities of the base station (from distance 5 to 0) the speed drops to $5/11 = 0.454$. The reason is that the distribution of neighbours for nodes around the base station is different from the distribution for distant nodes. More precisely, the nodes in close vicinity of the base station have very few nodes in the closer list but the number of nodes at the same distance or further away is high. The imbalance between the lists of neighbours grows with



(a) Mean number of hops

(b) Mean network bias

Figure 7: Perturbation impact on message delivery

the transmission range of the nodes, being more significant for the 20*neigh* configuration. In this case, the speed drops from 0.358 to 0.179 in the vicinity of the data sink.

As previously stated, the perturbation of the routing tables negatively impacts the efficiency of the data transmission protocol and thus affects the message delivery time.

We conducted a number of experiments for the same network configurations described before. We modified the routing tables of all the nodes using our perturbation algorithm, which is configured to perform at most 30 random swaps and uses input bias values between 0 and 1. For each simulation we sent 500 messages from 10 random source nodes located at the edge of the network, which is 20 hops away from the base station. The results are presented in Figure 7a, where the mean number of hops travelled by packets is depicted, and Figure 7b, which shows the relationship between the bias value used as input to the perturbation algorithm and the mean bias of the network after its application.

From Figure 7b we can observe that for those configurations with a larger number of neighbours, the range of values defined for the bias is smaller. This is the reason why the mean number of hops increases more abruptly as the bias approaches zero in configurations with fewer neighbours. In particular, when the desired bias is exactly zero, the mean number of hops for the 4*neigh* configuration is significantly high (over 1800 hops) because the mean network bias is slightly below zero (-0.0097). On the other hand, the mean hop count for the 20*neigh* configuration is below 350 hops because the mean network bias is close to 0.1.

In general, setting the desired bias value over 0.2 ensures that the mean number of hops for any configuration is below 100 for a source node located at the edge of the network.

6.3. Fake Traffic Overhead

The injection of fake traffic is a fundamental feature of the HISP-NC data transmission protocol since it hides the flow of real messages. However, the amount of fake traffic must be kept as low as possible in order to extend the lifetime of the nodes. To control the propagation of fake messages, our protocol defines a system parameter, $FAKE\_TTL$, that depends on the hearing range of the adversary in such a way that he is unable to observe the whole fake path. The idea is to prevent the adversary from controlling the transmissions of the node from which the first fake packet originated and the node which dropped the last fake packet, simultaneously. Otherwise, the attacker could learn information about the direction towards the base station.

Instead of injecting fake packets at regular intervals, which would provide the best privacy protection but would also deplete the sensors' batteries rapidly, the transmission of fake traffic is triggered by the presence of real packets. When the eavesdropping range of the adversary is large, the energy cost associated with fake transmissions would be similar to making sensor nodes inject fake traffic at regular intervals with the difference being that fake packets would be injected only in the presence of events.

In Figure 8 we illustrate the overhead imposed by HISP-NC for different time-to-live values and the various network configurations considered. More precisely, we show the ratio of fake over real messages that is introduced to balance the transmissions in a band around the real path. When $FAKE\_TTL$ is set to zero, the ratio is 1 because each real packet is transmitted in conjunction with a single fake packet, which is no longer propagated. The ratio is not affected by different network topologies since the number of transmissions performed by the protocol is independent of the connectivity of the sensor nodes. As the time-to-live grows, the ratio increase is in the order of $\mathcal{O}(2^{n+1})$ where n is the hearing range of the adversary. In any case, given the adversarial model considered in this paper the overhead imposed by this approach is moderate.

Finally, note the overhead imposed by fake transmissions might be reduced by half if we introduce a slight modification. Instead of sending two packets upon the reception of traffic, we might send a single packet addressed

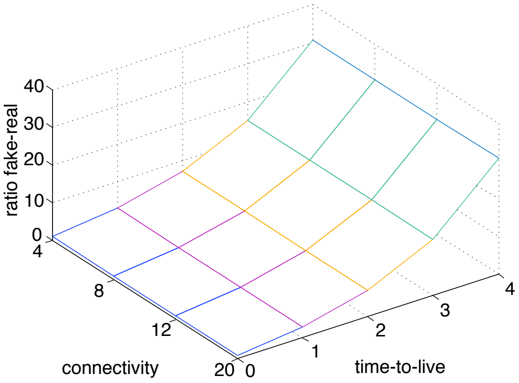


Figure 8: Overhead of fake messages

to two node identifiers. In this way, and assuming that the identifiers are hidden from potential observers, the two recipients receive the packet and continue with the forwarding process. The first identifier indicates the real recipient and the second indicates the fake recipient. This improvement is possible due to the broadcast nature of wireless transmissions, which allows all the neighbours of a node to overhear its messages.

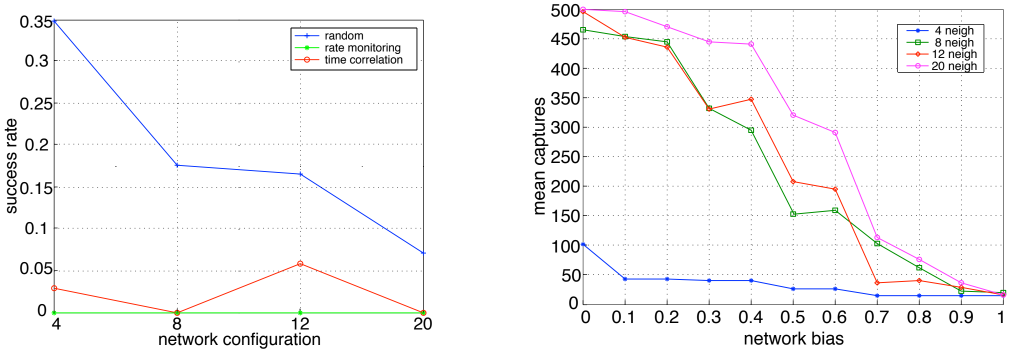
7. Privacy Evaluation

The HISP-NC data transmission protocol aims to provide protection from local adversaries capable of performing various types of traffic analysis attacks. The strategy of the adversary is to repeatedly move to a node closer to the base station by observing the transmissions along the communication path. Starting at any point of the network he eventually finds a data sender. From this location, the adversary attempts to determine the direction to the base station by observing the communications of the data sender and its neighbours.

Firstly, the adversary might perform a time-correlation attack and move in the direction of the neighbour forwarding the first message transmitted by the data sender. Given the features of our solution, several cases may occur depending on whether the packet is real or fake. If the packet is real, the adversary is highly likely to reduce by one, his distance to the base station. However, this is not necessarily the case because real traffic might also be forwarded in other directions. Moreover, the probability of following a real packet is lower than the probability of following a fake packet. The reason is that, as real messages move, they generate pairs of messages, one real and one fake, while fake messages trigger the transmission of pairs of fake messages. Also, note that the adversary can only be certain of whether he made the right choice when he follows a fake packet that is no longer propagated. This situation provides the adversary with no information about the direction to the base station because fake messages are forwarded in any direction. This is true unless the hearing range of the adversary allows him to observe both ends of the branch of fake messages. In that case, the adversary could determine that the root of the branch is closer to the base station with a high probability.

Alternatively, the adversary might choose to perform a sufficient number of observations before making a decision on the next move. In that case, the adversary will move towards the neighbour with the higher transmission rate. To reduce the success of this strategy, the HISP-NC transmission protocol makes nodes to evenly distribute messages among their neighbours, thus locally homogenising the number of packets being observed by a potential adversary. Again, the adversary cannot distinguish real from fake packets unless he observes a node which, after receiving a packet, does not forward it. This implies that he is at the edge of the band of fake messages surrounding the path of real data. Being able to precisely determine the limits of the band of fake messages could provide the adversary with information on how to reach the base station. However, the number and behaviour of events being reported by the sensor nodes may be extremely dynamic, which hinders the process of bounding the aforementioned band. Moreover, real packets are sent following a random walk which causes the band to be rather arbitrary. Consequently, even if the adversary was capable of delimiting the edges of the band at some point, this information does not necessarily lead him to the base station.

Notwithstanding, in an attempt to empirically demonstrate the validity of our privacy-preserving data transmission protocol we have launched a number of simulations with different types of adversaries starting next to the data sources, located at various distances from the base station ranging from 5 to 20 hops. Each experiment was executed for 500 simulation steps and we considered the same network configurations as in Section 7. First we ran simulations under a random adversarial model that, for each simulation step, moves to a random neighbour regardless of the transmission of messages. Then, we run the experiments with attackers performing rate-monitoring and time-correlation attacks. The results are depicted in Figure 9a.



(a) Traffic Analysis Attacks

(b) Node Capture Attacks

Figure 9: Success rate of different adversaries

We observe that the success rate of a random adversary is significantly higher than for the other two types of adversaries but still its success rate is close to or below 0.35. The random adversary is more effective for configurations where the average number of neighbours is smaller. Also note that in a quarter of the simulations, the adversary is placed only 5 hops away from the base station, which is when the adversary is more successful. Finally, it is worth noting that the success rate for the rate-monitoring adversary is zero at all times, however the time-correlation adversary reaches the base station occasionally, although the previous analysis suggests that this should not occur. The reason is that due to the nature of our simulator[4] we were unable to precisely represent the behaviour of a time-correlation adversary. Instead, the devised time-correlation adversary observes which messages are generated by the neighbours of the node and from those neighbours it randomly selects one as the next hop. The few times the adversary reaches the target is due to this random selection and because the initial position was only 5 hops away from the base station.

Additionally, we studied the success rate of an adversary performing node capture attacks. For each network configuration and bias value we ran 10 simulations, where the adversary started at random positions from the border of the network (i.e., 20 hops away). Again, each simulation consisted of 500 simulation steps, and we assumed that the adversary was capable of

[4] It is not possible to obtain the exact time at which a message is transmitted and thus sort messages based on their creation time.

capturing the routing tables of a node at each step. Also, we assumed that the adversary could move to the next node of interest to him by simply knowing its identifier but in a real setting the adversary might need to repeatedly capture neighbours until he eventually finds a particular node. Moreover, the adversary keeps track of the number of times he has visited each of the nodes in order to perform a more effective attack and prevent being trapped inside loops. Furthermore, the perturbation algorithm is configured to run during the deployment of the network for at most 30 iterations. Another parameter of the algorithm is the desired bias. However, if we used the same input bias for all nodes, provided that the distribution of the tables of the nodes might differ significantly, this would cause some nodes not to modify their routing tables at all and this issue could be exploited by the adversary. To prevent this, we adjusted the desired bias to the range of possible bias values of each particular node. In this way, the routing tables of all the nodes were perturbed to the same extent.

As expected, the number of captures an adversary needs to perform before reaching the base station increases as the bias of the network approaches zero (see Figure 9b). Clearly, the protection is more effective for configurations with a larger number of nodes[5] since the adversary keeps a record of already visited nodes and his strategy is to move to the first node in the routing table with the least number of visits. Although setting a very low bias is beneficial for protection against routing table inspection attacks it also negatively affects the delivery time of packets to the base station. Additionally, the number of tables an adversary might capture is rather limited due to the complexity of performing node capture attacks and also because compromising many nodes might reveal that the network is being attacked. In particular, if we consider that an adversary could capture at most a tenth of the nodes in the field, it is safe to use a bias value less than or equal to 0.5. Consequently, the bias is an important parameter that should be carefully tuned in order to find the right balance between usability and protection, based on the likelihood of node capture attacks.

However, it is worth noting that any attacker that is able to capture a node can behave as the node. Such an adversary has access to the (per-

[5] The number of nodes are 400, 1600, 1600, and 3600 for the configurations of 4, 8, 12, and 20 neighbours, respectively. Still, the distance from the edge to the base station is 20 hops in all cases.

turbed) routing tables and he can simulate the algorithm of the node, and by repeating this process all along the path, he will eventually reach the base station. This is true for any algorithm, not a problem solely of our solution, as long as the attacker can capture the routing tables and knows how the node works (i.e., he has all the secrets), he is able to simulate the nodes he is compromising. Still, implementing a perturbation algorithm is much better than not modifying the routing tables. In the latter case, the adversary simply needs to always move to the first neighbour in the routing table and he will reach the base station with the minimum number of steps.

8. Conclusion

This paper has presented HISP-NC, a novel receiver-location privacy scheme for WSNs. The proposed solution consists of a data transmission protocol and a routing table perturbation scheme that aim to prevent both traffic analysis and routing tables inspection attacks. The data transmission protocol relies on the injection of fake traffic to hide the flow of real traffic which is sent to the base station using a biased random walk. The goal is to probabilistically homogenise the overall number of packets that a node distributes among its neighbours while preserving three critical properties (i.e., convergence, homogeneity, and exclusion). These properties together, ensure the usability and robustness of the protocol against local adversaries. Additionally, we have defined the concept of routing table bias and based on this concept we have devised the HISP-NC perturbation scheme. This part of the solution consists of an optimisation algorithm that modifies the routing tables of the nodes to hinder inspection attacks while ensuring the delivery of data packets to the base station.

The feasibility of the HISP-NC scheme has been validated both analytically and experimentally through extensive simulations. In particular, we have analysed the impact of the connectivity of the network on the convergence of the data packets and the privacy protection level. Also, we have investigated the expected convergence time of packets in order to gain insights into the expected delivery delay of our solution. Moreover, we have explored the overhead imposed in terms of fake traffic injection for adversaries with different eavesdropping capabilities. Finally, we have discussed and evaluated the privacy protection achieved against adversaries performing different types of traffic analysis and node capture attacks. The proposed solution has proven to be secure and efficient against local adversaries capable of capturing a limited number of nodes in the network.

As for future work we are considering investigating new ways of reducing the fake traffic overhead required to protect against adversaries with a large hearing range. Also, we will explore the robustness of our scheme against more skilled adversaries. To this end, we first need to define a set of strategies based on the knowledge the adversary has about the network and the privacy protection protocol in use. The adversary may change his strategy depending on the context of the network. Countering such powerful adversaries may also require the development of new and more sophisticated protection mechanisms that have yet to be considered. Additionally, we are working on the design of a privacy-friendly topology discovery protocol since traditional solutions leak the location of the base station. Our final goal is to develop an integral solution capable of providing protection against attackers interested in both finding the base station and the data sources.

Acknowledgements

This research has been partially funded by the European Commission through the NESSoS (FP7 256890) project. Additionally, Ruben Rios and Javier Lopez have been supported by the PISCIS (P10-TIC-06334) and PERSIST (TIN2013-41739-R) projects funded by the Andalusian Government and the Spanish Ministry of Economy and Competitiveness, respectively. Also, Jorge Cuellar has received support from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no 609094 (RERUM).

References

[1] MATLAB - The Language of Technical Computing, http://www.mathworks.com/products/matlab/ (2014).

[2] U. Acharya, M. Younis, Increasing base-station anonymity in wireless sensor networks, Ad Hoc Networks 8 (8) (2010) 791–809.

[3] B. Alomair, A. Clark, J. Cuellar, R. Poovendran, Towards a Statistical Framework for Source Anonymity in Sensor Networks, IEEE Transactions on Mobile Computing 12 (2) (2012) 248 – 260.

[4] A. Becher, Z. Benenson, M. Dornseif, Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks, in: J. Clark, R. Paige, F. Polack, P. Brooke (eds.), Security in Pervasive Computing, vol. 3934 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2006, pp. 104–118.

[5] S. Chang, Y. Qi, H. Zhu, M. Dong, K. Ota, Maelstrom: Receiver-Location Preserving in Wireless Sensor Networks, in: Wireless Algorithms, Systems, and Applications, vol. 6843 of LNCS, Springer, 2011, pp. 190–201.

[6] J. Chen, H. Zhang, B. Fang, X. Du, L. Yin, X. Yu, Towards efficient anonymous communications in sensor networks, in: IEEE Global Telecommunications Conference (GLOBECOM), IEEE Communications Society, 2011, pp. 1–5.

[7] X. Chen, K. Makki, K. Yen, N. Pissinou, Node Compromise Modeling and its Applications in Sensor Networks, in: 12th IEEE Symposium on Computers and Communications (ISCC 2007)., 2007, pp. 575 –582.

[8] J. Deng, R. Han, S. Mishra, Countermeasures Against Traffic Analysis Attacks in Wireless Sensor Networks, in: 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM '05), 2005, pp. 113–126.

[9] J. Deng, R. Han, S. Mishra, Decorrelating wireless sensor network traffic to inhibit traffic analysis attacks, Pervasive and Mobile Computing 2 (2) (2006) 159–186.

[10] R. Di Pietro, A. Viejo, Location privacy and resilience in wireless sensor networks querying, Comput. Commun. 34 (3) (2011) 515–523.
URL http://dx.doi.org/10.1016/j.comcom.2010.05.014

[11] A. Eiben, J. Smith, Introduction to Evolutionary Computing, Natural Computing, 2nd ed., Springer, 2007.

[12] C. Gómez, J. Paradells, J. E. Caballero, Sensors Everywhere: Wireless Network Technologies and Solutions, Fundación Vodafone España, 2010, ISBN 978-84-934740-5-8.

[13] Y. Jian, S. Chen, Z. Zhang, L. Zhang, Protecting receiver-location privacy in wireless sensor networks, in: 26th IEEE International Conference on Computer Communications (INFOCOM 2007), 2007, pp. 1955–1963.

[14] Y. Jian, S. Chen, Z. Zhang, L. Zhang, A novel scheme for protecting receiver's location privacy in wireless sensor networks, Wireless Communications, IEEE Transactions on 7 (10) (2008) 3769–3779.

[15] R. Latif, M. Hussain, Hardware-Based Random Number Generation in Wireless Sensor Networks, in: Advances in Information Security and Assurance, vol. 5576 of LNCS, Springer, 2009, pp. 732–740.

[16] Y. Li, J. Ren, Preserving Source-Location Privacy in Wireless Sensor Networks, in: 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '09), 2009, pp. 493–501.

[17] G. Lo Re, F. Milazzo, M. Ortolani, Secure random number generation in wireless sensor networks, in: Proceedings of the 4th international conference on Security of information and networks, SIN '11, ACM, New York, NY, USA, 2011, pp. 175–182.

[18] K. Mehta, D. Liu, M. Wright, Location Privacy in Sensor Networks Against a Global Eavesdropper, in: IEEE International Conference on Network Protocols (ICNP '07), 2007, pp. 314–323.

[19] C. Ozturk, Y. Zhang, W. Trappe, Source-Location Privacy in Energy-Constrained Sensor Network Routing, in: 2nd ACM workshop on Security of ad hoc and sensor networks (SASN '04), 2004, pp. 88–93.

[20] S. Pai, S. Bermudez, S. Wicker, M. Meingast, T. Roosta, S. Sastry, D. Mulligan, Transactional Confidentiality in Sensor Networks, IEEE Security & Privacy 6 (4) (2008) 28–35.

[21] R. Rios, J. Cuellar, J. Lopez, Robust Probabilistic Fake Packet Injection for Receiver-Location Privacy in WSN, in: M. Y. S. Foresti, F. Martinelli (eds.), 17th European Symposium on Research in Computer Security (ESORICS 2012), vol. 7459 of LNCS, Springer, Springer, Pisa, Italy, 2012, pp. 163–180.

[22] R. Rios, J. Lopez, Analysis of Location Privacy Solutions in Wireless Sensor Networks, IET Communications 5 (2011) 2518 – 2532.

[23] R. Rios, J. Lopez, Exploiting Context-Awareness to Enhance Source-Location Privacy in Wireless Sensor Networks, The Computer Journal 54 (10) (2011) 1603–1615.

[24] C. E. Shannon, Communication Theory of Secrecy Systems, Bell System Technical Journal 28 (1949) 656–715.
URL http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf

[25] M. Shao, Y. Yang, S. Zhu, G. Cao, Towards Statistically Strong Source Anonymity for Sensor Networks, in: 27th IEEE Conference on Computer Communications (INFOCOM 2008), 2008, pp. 466–474.

[26] T. M. Vu, R. Safavi-Naini, C. Williamson, Securing wireless sensor networks against large-scale node capture attacks, in: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ASIACCS '10, ACM, New York, NY, USA, 2010, pp. 112–123.

[27] J. Walters, Z. Liang, W. Shi, V. Chaudhary, Security in Distributed, Grid, and Pervasive Computing, chap. Wireless Sensor Network Security: A Survey, Auerbach Pub, 2007, pp. 367–409.

[28] H. Wang, B. Sheng, Q. Li, Privacy-aware routing in sensor networks, Computer Networks 53 (9) (2009) 1512–1529.

[29] Y. Yang, M. Shao, S. Zhu, B. Urgaonkar, G. Cao, Towards Event Source Unobservability with Minimum Network Traffic in Sensor Networks, in: 1st ACM conference on Wireless network security (WiSec '08), 2008, pp. 77–88.

[30] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Computer Networks 52 (12) (2008) 2292 – 2330.

[31] B. Ying, J. R. Gallardo, D. Makrakis, H. T. Mouftah, Concealing of the Sink Location in WSNs by Artificially Homogenizing Traffic Intensity, in: 1st International Workshop on Security in Computers, Networking and Communications, 2011, pp. 1005–1010.

[32] L. Zhang, H. Zhang, M. Conti, R. Pietro, S. Jajodia, L. Mancini, Preserving privacy against external and internal threats in WSN data aggregation, Telecommunication Systems 52 (4) (2013) 2163–2176.