University of Malaga

Protecting Contextual Information in WSNs: Source- and Receiver-Location Privacy Solutions

by

Rubén Ríos del Pozo

submitted in fullfilment of the requirements for the
Degree of Doctor in Computer Science

Advisors

Fco. Javier López Muñoz

Full Professor at the University of Malaga

Jorge Ricardo Cuéllar Jaramillo

Principal Consultant at Siemens AG

November 2014

AUTOR: Rubén Ríos del Pozo

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

D. Fco. Javier López Muñoz, Catedrático de Universidad del área de Ingeniería Telemática del Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, y D. Jorge Ricardo Cuéllar Jaramillo, Investigador Principal en el Centro Tecnológico de Investigación en el Departamento CT RTC ITS de Siemens, Múnich,

CERTIFICAN QUE:

D. Rubén Ríos del Pozo, Ingeniero en Informática, ha realizado en el Departamento de Lenguajes y Ciencias de la Computación de la Universidad de Málaga, bajo nuestra dirección, el trabajo de investigación correspondiente a su Tesis Doctoral, titulada:

Protecting Contextual Information in WSNs: Source- and Receiver-Location Privacy Solutions

Revisado el presente trabajo, estimamos que puede ser presentado al tribunal que ha de juzgarlo, y autorizamos la presentación de esta Tesis Doctoral en la Universidad de Málaga.

Málaga, a 16 de Octubre de 2014

Fdo: Fco. Javier López Muñoz

Catedrático de Universidad del
área de Ingeniería Telemática

Fdo: Jorge Ricardo Cuéllar Jaramillo

Investigador Principal en Siemens AG,
CT RTC ITS, Múnich

"The heart of communication intelligence is not cryptanalysis but traffic analysis."

Privacy on the Line
W. Diffie and S. Landau

For all those who has suffered this thesis,
especially for my parents, wife, and daughter.

Acknowledgements

First of all, I would like to express my deepest gratitude to Javier Lopez, who believed in me even when I made a detour before I started to collaborate with him. Javier is not only a great professional, he is also a wonderful person and a dear friend. Thank you for letting me be part of your team and for understanding the situation after Lola was born. Also, special thanks to my second supervisor, Jorge Cuellar. He welcomed me in Munich and treated me as family, which is something I will be always grateful for. Thanks for showing me how beautiful research is when combined with maths and lots of “ají”.

During the development of this thesis I have met many people who have illuminated me in various ways. I have many memories about the moments I have spent with the guys at NICS Lab, you are awesome people that anyone would love to work with. There are many things I could thank you for but I will try to be serious. Onieva, thank you for deceiving me into pursuing a PhD. Thank you Carmen for your taps on my shoulder and putting up with my jokes. Isaac, thank you for convincing us of the benefits of QR-codes and BLE. Pepe you put samba and birdseed in our lives. Pablo thanks for your Amazon Prime account. Cristina thanks for the spoilers. Rodrigo you gave a chinese flavour to the lab. Ana, thanks for teaching us how to survive with potatoes and tuna. Thank you Fran for showing us a world of tags. David thanks for being our mexican guy. Lorena thank you for warning us about the hazards of gluten. Noelia thanks for making bureaucracy a bit easier. Gerardo thanks for the hacker support. Thank you Jesús for your silent giggles. Miguel thank you for the opposite. Dani thanks for your silly jokes and rhymes. Ángel thanks for coming by for an evening tea. Edu thanks for telling me I was always working on the same screen. Thanks for teaching us how to cook Spanish omelette with potato crisps, Troya. Saúl thanks for drinking all our water. Monte you helped to push stress out with your racket. Andrés thanks for being an artist. Thanks to all of you for making me enjoy at the office.

My long-life friends have been an endless source of inspiration, support, happiness and parties. To all my childhood friends (Fran, Chipi, Dani, Alvaro, . . .) and all the friends I have made at University (Carlos, Jose, Alejo, Juan Andrés, Kino, Juan Pablo, . . .) thank you all for being part of my life and for believing in me. My family deserves much more gratitude than I could ever write in a few lines. Thank you mum for being always there for whatever we need. Thank you dad for being a model to me. Thank you Sara for being always proud of your little brother. Thank you Paco for being a friend and a brother. Milena and Lola, you are my life and I love you more than I could ever have imagined. Without you this thesis might still have been possible, possibly two years earlier, but my life would have not been complete.

I am also grateful to some institutions and projects for their funding and support. This thesis has been primarily funded by the Spanish Ministry of Education under the FPU fellowship. Special thanks goes to the NESSoS (FP7 256890) and ARES (CSD2007-00004) projects because they have given me the opportunity to collaborate with other institutions, receive feedback for improving the quality of my research, and make good friends.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the research domain, problems and contributions of this thesis. First, it introduces the concept of Wireless Sensor Networks, especially highlighting their most distinguishing features, which differentiate them from other computing paradigms and makes them attractive for use in countless application scenarios. This chapter also shows how these particular features affect, to a considerable extent, the security and privacy of the network and the environment being monitored. Subsequently, the various sorts of privacy problems arising from the deployment and use of Wireless Sensor Networks are described and then the chapter concentrates on the privacy issues arising from the analysis of the traffic generated by these networks. The main problem addressed in this thesis, i.e., the location privacy problem, is then considered. This problem is motivated with a critical scenario where safeguarding location information is of paramount importance for the safety and survivability of both the network and the entities being monitored. Finally, this chapter ends with a list of the main contributions underpinning this thesis and a brief description of its research goals and the outline of the following chapters.

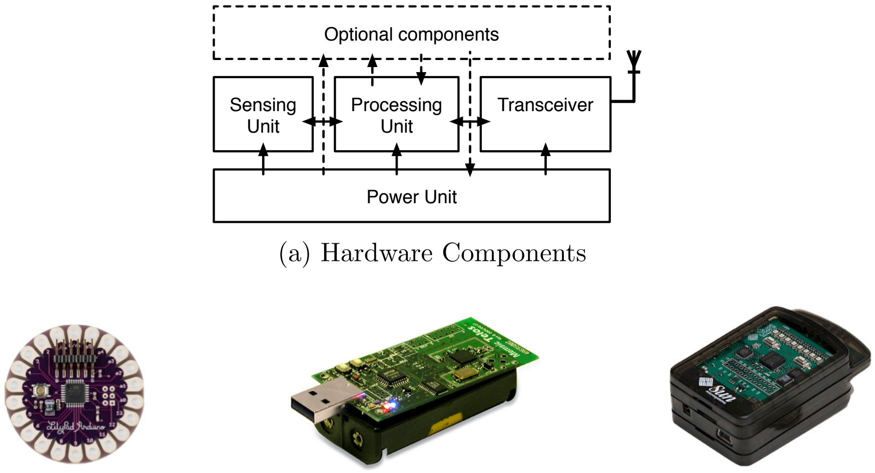
1.1 Wireless Sensor Networks

In the near future the world is expected to be teeming with a huge amount of smart and interacting objects offering potentially tremendous opportunities both to industry and final users. In a foreseeable scenario such a this, already known as the Internet of Things [137], everyday objects will be fitted with computational and sensing capabilities. This thesis concentrates on one of its supporting technologies, Wireless Sensor Networks, and some of the privacy implications arising from their deployment in different scenarios for monitoring purposes.

A Wireless Sensor Network (WSN) [49] is a distributed network consisting of two types of devices, namely, the sensor nodes and the base stations. The sensor nodes (or motes) are matchbox-sized computers which have the ability to monitor the physical phenomena occurring in their vicinity and to wirelessly communicate with devices nearby. To the contrary, the base station (or sink) is a powerful device, usually a computer, that collects and processes all the information collected by the nodes. The base station serves as an interface between the sensor nodes and the users. In some sense, sensor nodes extend the capabilities of ordinary computers by allowing them to feel and interact with the world around them. In this metaphoric view of WSNs [123], the sensor nodes represent the sensory cells, the wireless channels behave as the nerves, and the base station can be regarded as the brain of a living organism.

Sensor nodes can be fitted with a large variety of physical sensors (e.g., temperature, presence, vibration, radiation, etc.), which makes of WSNs a highly customisable technology capable of performing many diverse tasks. The versatility of the devices combined with their small size permit sensor networks to be unobtrusively embedded into systems for the purpose of monitoring and controlling very diverse environments and assets. In fact, WSNs have been successfully applied to precision agriculture and farming [66, 82], habitat and environmental monitoring [78, 85, 87], e-health and assisted-living control [33, 147], industrial control and critical infrastructure protection [19, 45], structural health monitoring [23, 65], homeland security and military applications [47, 51], among many others.

One of the most distinguishing features of WSNs is the severe hardware limitations of sensor nodes. From an architectural point of view, these tiny devices consist of four essential components [3]: the sensing unit, the processing unit, the transceiver, and the power unit (see Figure 1.1a). The sensing unit consists of a series of physical sensors which provides the node with the ability to measure different environmental conditions, as previously stated. The processing unit consists of a simple micro-controller whose computational capabilities and memory space are limited to a few Megahertz (typically between 8 and 16Mhz) and a few kilobytes (typically between 4 and 10kB for RAM memory, and between 48 and



(a) Hardware Components

(b) LilyPad [11] (c) TelosB [91] (d) SunSPOT [96]

Figure 1.1: Wireless Sensor Nodes

128kB for instruction memory), respectively. Although this is the most typical configuration for sensor nodes, these values may vary depending on the application scenario they are intended for. Therefore, we can distinguish three classes of sensor nodes: extremely constrained sensor nodes, typical sensor nodes, and high-performance sensor nodes. Figure 1.1 shows one example of a sensor node for each of these categories.

The transceiver allows the sensor node to send and receive messages wirelessly at a low data rate (between 70 and 250kB/s) usually in the 2.4 Ghz spectrum. Furthermore, the maximum communication range outdoors is around 100 meters for low-power configurations. The power unit is in charge of supplying energy to all the components. Finally, note that the sensor node might be fitted with some optional components depending on the application's requirements. These components include, but are not limited to, localisation systems (e.g., GPS chips), power scavengers (e.g., solar panels), mobilizers, and external flash memories.

The power unit is, in fact, the most limiting factor in sensor nodes because all other components depend on it to operate. Since the power unit usually consists of two AA batteries (i.e., 3 volts), which cannot be replaced or recharged once the network has been deployed, all other components must use energy responsibly. This issue has several important implications, for example in the operating speed of the nodes. The most common built-in micro-controllers can operate at different speeds [12, 134] but due to the limited voltage supplied by the batteries, by default the operating system of the sensor nodes reduces the speed to the minimum. Also, it is well known that the energy consumed by the transceiver is far greater than the energy consumed by the micro-controller [132][1] and thereby it is advisable to favour computation over transmissions.

The communication model in WSNs is clearly affected by the previous features. Although the data reporting methods in WSNs can be either time-driven, query-driven, or event-driven, the latter is the most usual mainly due to the fact that it is suitable for time-critical applications as well as being energy efficient. In the event-driven model, a sensor node starts reporting data to the base station immediately after an event of interest (i.e., a (sudden) change to the properties of a particular phenomenon) has been detected in its vicinity and stays silent otherwise. Consequently, if there are no events to be reported, the energy consumption of the nodes is moderately low. Moreover, since the transmission power of the nodes is usually not sufficient (or too energy consuming) to establish a direct communication with the base station, the data source uses multi-hop communications to deliver the sensed data.

Many routing protocols have been devised to allow remote sensor data to reach the base station [4]. However, there are two main approaches that stand out from others, namely flooding-based and single-path routing protocols. A baseline flooding is a simple routing algorithm in which every incoming message is transmitted to all its neighbours but the one which sent it[2]. Recipient nodes repeat the process thereby making the packet eventually visit all the nodes in the network. This approach is very reliable because it provides a lot of redundancy but it is also very energy inefficient because in a typical WSN the only intended recipient is the base station. On the contrary, a single-path (or shortest-path) routing protocol is intended to minimise the number of relaying nodes to reach the destination of the message. In a single-path routing, whenever a node has event data to transmit, it sends a message to a neighbouring node which is closer[3] to the base station than itself. This process is repeated for each of the nodes in the communication path until the data is eventually delivered to the base station, as depicted in Figure 1.2. All future messages from the same source node will

[1] The analysis is based on the Mica2 sensor platform, which uses an Atmel128L micro-controller and a CC1100 chip transceiver. When active, the micro-controller consumes 8mA while the radio consumes 7mA in listening mode and up to 21.5mA for maximum transmission power (+10dB).

[2] Actually, every neighbour receives the message but the original sender discards the message.

[3] The distance is usually measured in terms of the number of hops (i.e., intermediaries) that are necessary to reach a particular node.

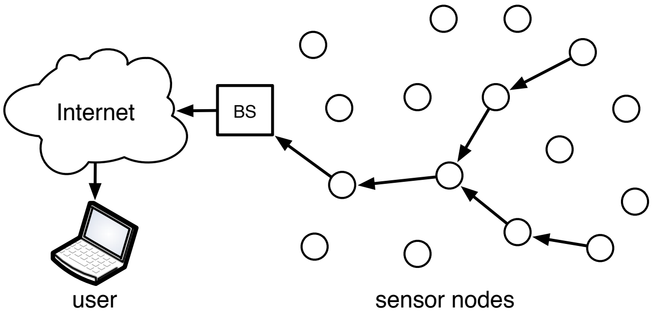


Figure 1.2: Communication model in WSNs

follow the same path unless a topology change (e.g., due to the death of a node) occurs.

Additionally, some sensor networks may take advantage of data aggregation protocols to further reduce network traffic on its way to the base station. Data aggregation consists of a set of operations (e.g., counting, average, maximum, minimum) that are performed at some intermediate points of the network to combine the data originating from different sources.

1.2 Overview of Security in Wireless Sensor Networks

Despite the unprecedented benefits that WSNs bring to our society, there are many relevant issues that demand meticulous attention. Sensor networks are inherently insecure and the severe hardware limitation of sensor nodes requires most well-founded security solutions, which can effectively protect traditional computer networks, have to be adapted or they simply do not work on these devices. Additionally, the unattended nature of these networks (i.e., once deployed, usually in remote or hostile environments, no network maintenance is done) provides an attacker physical access to the devices, which are rarely tamper-resistant due to the implications in the overall cost of the network. Furthermore, the broadcast nature of the transmission medium gives access to the packets exchanged by the sensor nodes to anyone within the communication range.

Consequently, adversaries may take advantage of the distinguishing features of WSNs in order to launch attacks against the network. Sensor networks are particularly vulnerable to attackers who may hinder the correct operation of the network and thus annul all the potential benefits of this technology. Based on their capabilities, adversaries can be classified as [142]:

- *Internal – external*: the distinction between an internal or an external attacker lies in whether the attacker is a member of the network or an outsider. External attacks are performed by entities which do not belong to the network, while internal attacks are performed by legitimate nodes which behave in an unintended way.
- *Passive – active*: the distinguishing feature between these attackers resides in their ability to disrupt the normal network operation. A passive attacker is an eavesdropper and limits his actions to merely observing the messages exchanged by the sensor nodes. In contrast, the active attacker does not only listen but may introduce new packets, modify or block packets in transit, tamper with the devices, or a combination of these.
- *Mote-class – laptop-class*: the main difference between these attackers is on their hardware resources. A mote-class adversary has capabilities similar to an ordinary sensor node while the laptop-class adversary can use powerful devices with greater transmission range, processing power, memory storage, and energy budget than typical sensor nodes. A similar distinction is made between local and global attackers when referring to the ability of the attacker to control only a part or the whole network.

The most challenging security-related task in WSNs is to maintain the availability of the network due to the constrained and unattended nature of the nodes, which cannot do much to protect themselves. A powerful adversary can easily target the devices themselves, the batteries, or the communication channels, in order to disrupt the network operation. Nonetheless, there are some other threats which do not necessarily affect the availability but rather the confidentiality and integrity of the communications or the system as a whole. Some of these attacks may affect several security properties at once. Below we provide a non-exhaustive list of potential threats affecting WSNs [139]:

- *Denial of service attacks* are any action that reduces or neutralises the ability of a device or network to perform as expected. A standard DoS attack is jamming, which consists of the transmission of a signal that interferes with and affects the frequency spectrum used by the sensor nodes. Similar attacks may also be launched at the link layer by generating collisions, at the routing layer by dropping packets, or at the application layer by flooding a sensor node with so many requests that it eventually runs out of memory or battery.

- *Information flow attacks* target the communication channels in order to compromise the confidentiality and/or the integrity of the transmissions. An attacker may simply observe the communications but he may also intercept, modify, replay or fabricate messages. The first type of attack is intended to retrieve valuable information from the packets traversing the network while the remaining attacks mainly focus on deceiving the base station to accept a false data value or are part of a more sophisticated attack.

- *Physical attacks* are actions targeting the hardware components of the sensor nodes. In this type of attacks the adversary has physical access to the sensor node and may access any information contained in it, such as the event data, the program binaries, or the cryptographic material. Additionally, the attacker may modify information within the sensor node in order to create a compromised version which is under the control of the adversary. This is usually the first step in performing the identity attacks described below. Furthermore, the attacker may change the topology of the network by moving some sensor nodes to a different network position, removing them or destroying the devices.

- *Identity attacks* concentrate on spoofing or replicating the identities of legitimate nodes of the network. The Sybil attack is a form of identity attack where a single node uses multiple (new or stolen) identities. In contrast, a node replication attack consists of having several nodes in the network using the same identity. The final goal of these attacks depends on the application but they are usually effective in obstructing routing algorithms, intrusion detection and any voting-based mechanisms.

- *Protocol attacks* concentrate on disrupting routing protocols, data aggregation mechanisms and other platform-specific operations. Attacks on the routing layer include attracting or deviating network traffic from particular regions, increasing latency, or dropping messages. In a selective forwarding attack a malicious node drops some of the packets it receives based on a given criteria. A sinkhole attack aims to attract network traffic towards a particular node controlled by the adversary. In a wormhole attack, a compromised node receives packets and tunnels them (e.g., using a directional wireless link) to another point of the network, and then replays them back into the network. This behaviour may disrupt some neighbours discovery mechanisms. Additional attacks include the injection of fake control packets (i.e., hello flood and acknowledgement spoofing attacks).

Consequently, the need for security mechanisms in WSNs is undeniable and key management schemes are essential. The goal of key management schemes [158] is to generate, distribute, update, and revoke the key material necessary to establish secure (i.e., confidential, unforgeable, and authenticated) communication channels between nodes. Many of the previous attacks can be countered by this means, however, implementing robust and efficient security primitives in WSNs, especially public key cryptography, is very challenging given the resource constraints of sensor nodes.

Nonetheless, the use of cryptography is not enough to protect the nodes from physical attacks. To that end, the hardware layer could integrate some protection mechanisms, like tamper-resistant modules, but these would significantly increase the cost of each individual node. A more affordable solution is to use code obfuscation and data scrambling mechanisms [6], which would slow down and possibly prevent the analysis of the internals of the nodes. Additionally, the sensor network may incorporate trust management systems [80] and intrusion detection systems [9] to further protect the network.

1.3 Privacy in Wireless Sensor Networks

Extensive work has been done on the protection of WSNs from the hardware to the application layer, however privacy preservation has not received that much attention in these scenarios. Before sensor networks are pushed to the forefront, it is absolutely necessary to consider and address all potential privacy risks that may arise from the adoption of this technology. As a matter of fact, advances in

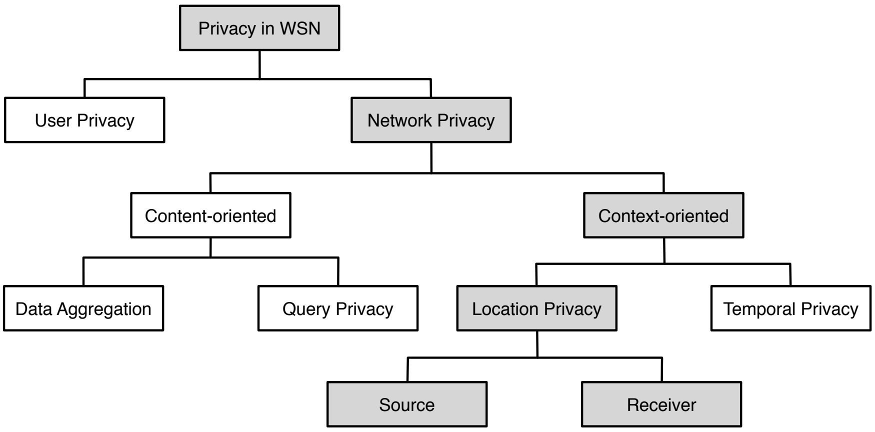


Figure 1.3: Classification of Privacy Problems in WSN

technology[4] have always revolutionised the way in which privacy is violated and protected.

We distinguish two major categories of privacy problems due to the deployment of WSNs. This taxonomy is based on the entity that is aiming to breach privacy and the entity whose privacy is violated. In the first group, the privacy perpetrator is the network (owner) itself while, in the second group, an external entity takes advantage of the network to obtain sensitive information. In either case, the information obtained by the privacy perpetrator may be related to individuals or other critical assets. As shown in Figure 1.3, the classification can be broken down into further categories for specialised problems. This dissertation concentrates on the part of the tree coloured in grey.

1.3.1 User privacy

The most obvious privacy risk is due to the unobtrusiveness and ubiquity of sensor nodes, which allows them to inadvertently spy on anyone or anything within the reach of the network. Moreover, the reduced cost and size of sensor nodes favour the deployment of large-scale surveillance networks, which may go unnoticed by unaware individuals. Furthermore, their ability of collaboratively analyse and automatically correlate data at different periods of time can result in highly accurate tracking and profiling applications.

[4] A very good paper by Jan Holvast [53] compiles a history of privacy. The author shows how the birth of new inventions and technologies (e.g., the printing press, automatic photography, the Internet, and so forth) have influenced and invaded the privacy of individuals and organisations.

As far back as 1991, Mark Weiser [145] warned of the importance of privacy protection in ubiquitous computing scenarios, where sensing technologies are one of the cornerstones of these environments:

> *"hundreds of computers in every room, all capable of sensing people near them and linked by high-speed networks, have the potential to make totalitarianism up to now seem like sheerest anarchy".*

This type of threat cannot be easily tackled by technological means alone; rather, severe laws, regulations, audits, and sanctions are also absolutely necessary to prevent ill-intentioned entities from invading individual privacy, since the benefits usually outweigh the consequences. In this case, privacy-friendly engineering approaches [133] are off-topic because the owner or administrator of the network is a malicious entity (e.g., a governmental agency) and is therefore unwilling to admit that the network is used for surveillance activities.

However, legitimate networks may opt to apply a privacy-by-policy approach thereby informing the user of the collection of personally identifiable information[5] and the application of fair information practices to these data. However, making the sensor network responsible for presenting privacy policies to the user in a meaningful and unobtrusive way is not a trivial task, especially due to a lack of adequate interfaces. Another option is to let the user define his/her own privacy preferences in order to illustrate how much privacy he/she is willing to give up when interacting with the network. Some approaches [67] have concentrated on these policy-agreement protocols to protect users privacy but in most cases if the policies do not agree, the user cannot access the service. Another limitation to these approaches is how to ensure that the policies are correctly defined and suitable for each user's privacy expectations.

A more suitable approach is to follow the privacy-by-architecture (or privacy-by-design) principle, which not only minimises the collection of personally identifiable information but also promotes client-side data storage and processing. As personal data only leaves the user domain after sufficient care has been taken to correctly anonymise and reduce the quality (e.g, by adding noise, reducing the precision, etc.) of the data in such a way that it is still useful for the provision of the service but it does not leak private information. An extensive body of

[5] Personally identifiable information [88] is any information that (a) can be used to distinguish or trace an individual's identity, and (b) is linkable to an individal.

research has concentrated on the disassociation of identity and location information [16, 20, 34, 50] because of the criticality of these data. Knowing the location of a person at a particular moment reveals a lot of information, especially if these data are periodically accessible. Therefore, if one observes that an individual is at a particular location at 3 am this might indicate that this individual is at home, but after continuous observations of the same location the initial hypothesis becomes much more plausible.

1.3.2 Network privacy

While it is undoubtedly true that the use of a WSN for surveillance purposes is detrimental to individual privacy, there are a number of issues that may affect the privacy of the network itself and, consequently, the privacy of the individuals and assets it monitors. And more importantly, these new privacy problems are still present despite the application of fair information practices.

Network privacy problems can be classified as content-oriented and context-oriented threats [100]. This categorisation is based on the type of data the adversary is interested in or is capable of retrieving from the sensor network.

Content-oriented privacy

Content-oriented privacy threats are mainly due to the ability of an adversary to observe or in some way manipulate the actual contents of the packets traversing the network. A clear example scenario is the Smart Grid [46], where smart meters (i.e., household embedded devices used for measuring utility consumption) use adjacent meters to relay consumption data to a readings collection device, which in turn transmits the readings to the utility company for the purpose of billing[6]. Clearly, the scenario described (see Figure 1.4) has many similarities to a typical WSN, where the smart meters behave as sensor nodes and the readings collection device is like a base station.

[6] The Smart Grid scenario is far more complex [55], as it also includes the generation, transmission, distribution, operation, and market domains.

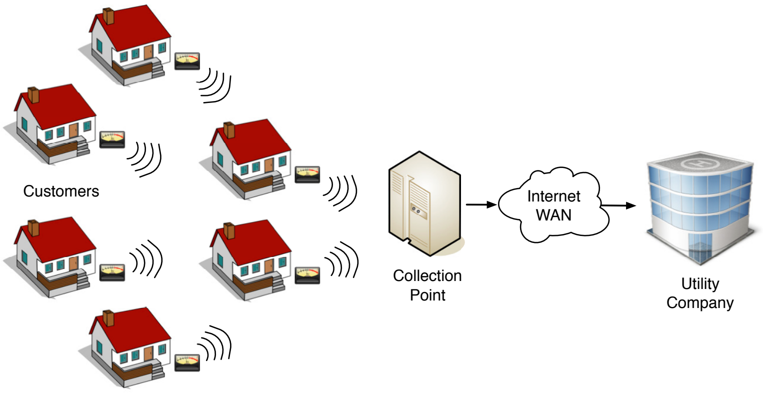


Figure 1.4: Simplified Smart Metering Scenario

A first line of defence to protect content-oriented privacy in these scenarios is to apply secure encryption schemes in order to provide confidentiality and integrity to the data in transit[7]. However, this straightforward countermeasure can only provide protection from a subset of potential adversaries, that is, external observers. An internal attacker (i.e., a legitimate sensor node controlled by a malicious entity) can still intercept, store, and analyse the data being broadcast by its neighbours since it owns legitimate decryption keys. To prevent intermediaries from peeking at the data of other nodes (e.g., the electricity consumption from a neighbour), it is possible to apply end-to-end encryption between the data source and the sink. This is a simple and effective solution but it presents at least two problems, namely, the need for additional key material to allow the destination to decrypt the messages probably without even knowing the original sender, and the disruption of some common operations performed by the network. In particular, end-to-end encryption precludes the use of data aggregation protocols because intermediate aggregator nodes are unable to combine their own data with that contained in the packets received.

Therefore, the research community has struggled to develop privacy-aware data aggregation mechanisms capable of preventing insider attacks. The main idea behind most of these solutions is to perturb the original data in such a way that an aggregator cannot obtain the contribution of a single source node even though the aggregated result remains correct. Some solutions are based on the

[7] Confidentiality and privacy are different yet related properties. Confidentiality does not necessarily imply privacy, it only prevents unauthorised access to data. Consequently, if the original data is personally sensitive, confidentiality helps to enforce privacy, while this is not the case the other way around.

addition of noise to the contributions of the sensor nodes in the form of random values that can later be removed by the aggregator [52]. Some other approaches leverage on the properties of homomorphic encryption schemes in order to allow intermediate nodes to aggregate their own data to received packets without the need to decrypt them. To allow decryption of aggregated data, each node shares a secret key with the base station [22] or uses multiple random keys from a network-wide key pool shared (or not) with the base station [159]. Additionally, some authors have provided solutions to some specific aggregation functions such as additive [52] and histogram [54, 160] operations. The former proposes slicing the data into chunks and distributing them to some neighbours, which finally add (i.e., aggregate) the shares received from all its neighbours before submitting the result to the base station. The latter uses a histogram of a particular granularity and each of the sensor nodes informs to the base station of not the real value but rather the histogram interval where its readings lie.

Finally, some research on protecting query privacy in WSNs has also been conducted. This problem refers to the ability of an attacker to determine the contents of a query sent to the network based on (the identities of) the nodes that respond to that particular query. Some schemes propose hiding the actual target node by also issuing bogus queries to other sensor nodes in the network [21]. A similar approach [37] is to issue a query following a particular path in the network such that the target node is potentially any node in the path. A simpler and more privacy-preserving solution is to query all sensor nodes in the network any time a user is interested in data from a particular node. However, this approach is unfeasible for densely populated sensor networks due to the huge amount of network traffic generated unless an efficient data-aggregation scheme [42] is used to reduce the amount of traffic and still provide a perfect privacy protection.

Context-oriented privacy

Context-oriented privacy concerns the protection of the data generated from the operation of the network. These data are not part of the actual packet contents exchanged by the sensor nodes, they are instead metadata associated with the measurement and transmission of the sensed data. Therefore, even if the payloads are suitably protected from eavesdropping, an adversary could obtain other sensitive information that might compromise the privacy of the network itself and the privacy of the events being monitored. In fact, the mere presence of messages traversing the network is usually indicative that some kind of event is taking place.

Traffic analysis [107] is a very powerful set of mechanisms that helps to determine information about the entities exchanging information by observing the attributes of the communications. Pai et al. [101] show how apparently innocuous (meta)data, obtained from the simple observation of network traffic, can be used to infer sensitive information about the network:

- The *frequency spectrum* used for the communication might reveal the sensor platform being used. Recent technologies (e.g. micaz, IRIS, Imote2) can be easily distinguished from older ones (e.g. cricket, mica2) as the former perform in the 2.4 GHz spectrum while the latter perform at sub-GHz frequencies. So, by using a spectrum analyser, an attacker might be able to determine the owner of the network since different frequency bands are assigned and licensed for different purposes and to distinct organisations. In addition, being able to distinguish the types of sensor nodes in use may allow an attacker to exploit platform-specific vulnerabilities.

- The *transmission rate* at which messages are generated and delivered to the base station is a good indicator of the quantity and nature of the events being monitored. In event-driven sensor networks, the transmission of messages reveals the presence of events in the network to an observer. Similarly, the absence of messages might be an indicator of sensitive information. Consider, for example, a sensor network deployed to monitor the heartbeat of a patient. A high transmission rate might indicate that the patient is in a stressful situation, while a low or a complete lack of messages may imply that the patient is sleeping, has fainted, or has suffered a cardiac arrest.

- The *message size* provides information about the type and precision of the data being collected. When a sensor network is used to monitor phenomena of different granularity (e.g., presence (boolean) and radiation (double)), the attacker can easily distinguish which type of event data is contained in each message based on its size. Additionally, the adversary can guess the purpose of the network given the deployment scenario and the message length because a coarse-grained data collection is used for slow-varying phenomena while a fine-grained data collection is suitable for fast-varying phenomena. Moreover, some data aggregation protocols might introduce privacy issues because as the nodes incorporate their own data to received messages, the messages increase in size. This feature can help the adversary determine the proximity to the base station since the lengthy packet has traversed many nodes.

- The *communication pattern* might reveal information about the network topology. Any solution for WSNs is especially tailored to preserve the limited battery of sensor nodes in order to extend the lifetime of the network. In particular, the event-driven data reporting model and the use of shortest-path routing protocols is intended to reduce the high cost associated with wireless communications. However, an adversary can exploit these features to discover the location of important network nodes, generally the data sources and the base station.

An additional contextual privacy consideration is made by Kamat et al. [61], who suggest that it is also important to hide the time of occurrence of events because this information may allow an adversary to predict future behaviours of the phenomenas being monitored by the network. This privacy problem, known as *temporal privacy*, is especially relevant in mobile asset monitoring applications since the adversary may guess the pattern of movement of these assets.

Admittedly, some of the problems introduced by these features can be circumvented by implementing simple countermeasures, like a fixed message size regardless of the length of the contents. However, concealing the information associated with some other features is less straightforward. Preventing the disclosure of location information about relevant network nodes is a particularly challenging and safety-critical task.

1.4 Location Privacy in Wireless Sensor Networks

Based on the original privacy definition by Alan F. Westin [5], location privacy can be defined as the desire to determine under what circumstances and to what extent location information is exposed to other entities. Therefore, location privacy in WSNs aims to preserve the location of relevant nodes in the network.

More precisely, it focuses on preventing an adversary from determining the location of the data sources and the base station. These are called respectively the source- and the receiver-location privacy problems.

1.4.1 Motivating Scenario

In order to illustrate the importance of location privacy problems in WSNs and to facilitate future analysis and discussion, we present a motivating scenario that captures the most distinguishing features of both source-location and receiver-location privacy. The criticality of the scenario highlights the importance of the problem and the need to develop solutions to protect from adversaries.

Consider a military environment like the one depicted in Figure 1.5, where a large number of sensor nodes are deployed in a vast area for the purpose of monitoring the movements and whereabouts of the troops and assets (e.g., armaments, tanks, drones, etc.) belonging to a military force. The goal of the network is to better coordinate and control the troops during attack and reconnaissance missions. Doubtlessly, the deployment of a monitoring sensor network can mean a significant advantage over the enemy.

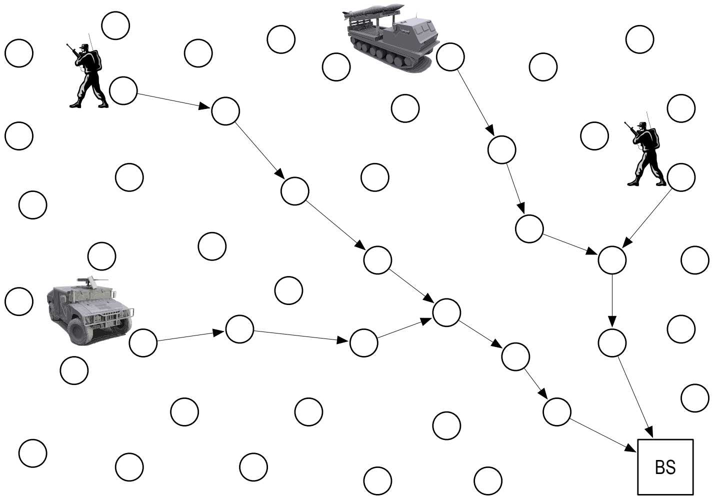


Figure 1.5: Sensor Network Deployment for Military Operations

Given the critical nature of the scenario, information must be processed and analysed in real time. Therefore, immediately after the detection of an event of interest (e.g., the presence of troops) in the area controlled by a sensor node, the collected information is transmitted towards the base station on a multi-hop basis. Typically, single-path or flooding-based routing algorithms are used. As long as the event persists, the corresponding sensor node will continue to generate new traffic, which is expected to reach the sink in the shortest time possible. In the meantime, an adversary (i.e., the enemy) will try to exploit the deployed infrastructure for his own benefit.

The importance of source-location privacy is not the protection of the hardware itself but on the need to hide to presence of events in the field. Especially sensitive scenarios are those involving individuals and valuable assets, like the military scenario depicted in Figure 1.5. An adversary who knows the location of a source node can determine with sufficient precision the area where an event has been detected, meaning that the enemy is capable of uncovering the location of targets in order to attack them. Moreover, protecting the location of the base station is extremely important because if it is compromised or even destroyed, the whole system is rendered useless. Besides the physical protection of the network, the location of the base station is strategically sensitive because this key device is most likely housed in a highly-relevant facility. In the military scenario under consideration, the attacker can accomplish a more devastating attack by targeting the base station, which is located within the headquarters of its enemy.

Despite the criticality of the military scenario, the location privacy problems are extensible to any conceivable scenario due to the singular communication pattern of WSNs. The attacker may exploit these properties to find the base station or data sources. To better understand the threat we must confront, it is necessary to know the general features of the adversarial model. The adversary is assumed to be aware of the methods and protocols being used by the network or he can eventually deduce them after sufficient observation of the network behaviour. In other words, the adversary is assumed to be *informed*. Normally, he does not interfere with the normal operation of the network so as not to be detected because the network may implement intrusion detection mechanisms that alert of abnormal situations. This would hinder the plans of the adversary or it could result in unwanted consequences (e.g., being attacked). Therefore, the attacker is considered to be *passive*. Additionally, the adversary has no control over the sensor network but in certain scenarios he may be able to capture and compromise some nodes to help him determine the location of particular nodes. So, the adversary is generally assumed to be *external*. Finally, depending on the power of the adversary, he may need to move in the field in order to find the target or he can remotely determine its location based on the analysis of the traffic

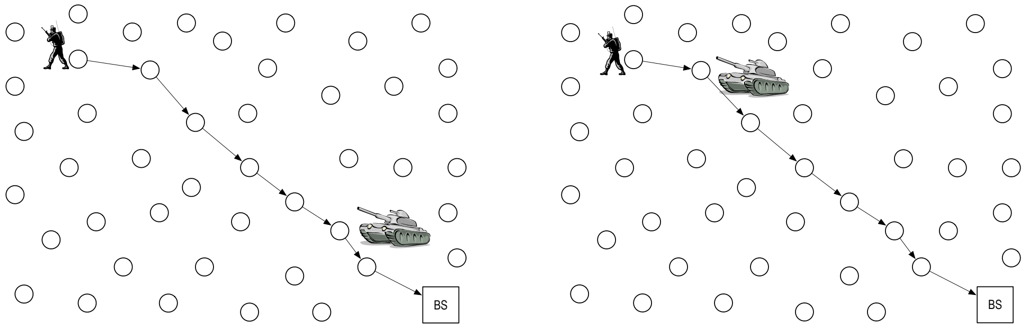


Figure 1.6: Source-location privacy problem in a military scenario

captured by an adversarial network deployed for the purpose of eavesdropping the communications of the legitimate network. With respect to the hearing range of the adversaries, they can be considered either local or global.

1.4.2 Source-Location Privacy

The source-location privacy problem was first introduced and analysed by Ozturk et al. [100]. The authors show how the operation of various routing protocols widely used in WSNs (i.e., single-path and flooding algorithms) leak information about the nodes reporting event data to the base station.

An adversary with a local vision of the network communications, namely a mote-class adversary, can act in the following way to find the source of event messages. Starting at any point of the network[8] and moving around, the attacker eventually stumbles upon a communication path originating from a remote sensor node. The adversary, who is equipped with a device capable of measuring the angle of arrival of received signals (i.e., a directional antenna), can estimate the sensor node which transmitted a message. This node is a mere intermediary in the communication path but by moving towards it and repeating the same process over and over again, the adversary can finally reach the original data source. This process is depicted in Figure 1.6, where the enemy (i.e., the tank) follows the communication path in reverse in order to find the soldiers. Thus, this strategy is usually referred to as traceback attack.

Note that the situation is not any better when there are multiple source nodes reporting event data to the base station. The reason is that the adversary is usually not interested in reaching a particular data source since all events are

[8] Usually the attacker is assumed to start in the vicinity of the base station from where he can observe any incomming communication.

equally important to him[9]. In the military scenario considered here, any data source guides the enemy to a target. Similarly, in an endangered animal monitoring scenario, the location of a source node leads to an animal. In a cargo tracking application, data sources are directly related to the location of the cargo, and so forth.

Some adversaries can achieve a global vision of the sensor network by deploying their own adversarial network. Therefore, the adversary does not need to move in the field, instead he can simply analyse the data collected by his network remotely. In this case, each adversarial node monitors the transmissions in its vicinity and, based on the number of packets overhead by each node, the adversary deduces the location of the data sources. In particular, the adversary can spot the area where a data source is located because sensor nodes only initiate a transmission in the presence of events. Moreover, the time at which a transmission takes place helps to determine the location of the source node. Clearly, the attacker can spot data sources by comparing the time at which any pair of adversarial nodes first observed a sequence of messages.

Additionally, some adversaries might also compromise a small portion of the sensor nodes in the network in an attempt to obtain information about the data sources. Since data is transmitted using multi-hop routing mechanisms, an adversary compromising a portion of the network has a certain probability that some of the nodes he controls is involved in the routing of the data to the base station. As compromised nodes are part of the network they have access to the any secrets shared with neighbouring nodes and thus they could access the contents of the messages it forwards. Having access to the packet contents may allow en route nodes to retrieve the original data sender because this information must be contained somewhere in the packets to allow the base station recognise the data source.

1.4.3 Receiver-Location Privacy

The base station is the most precious element in a WSN and, as such, its location must be thoroughly protected from potential attackers. Deng et al. [39] started to investigate along this line by presenting a set of mechanisms that included the

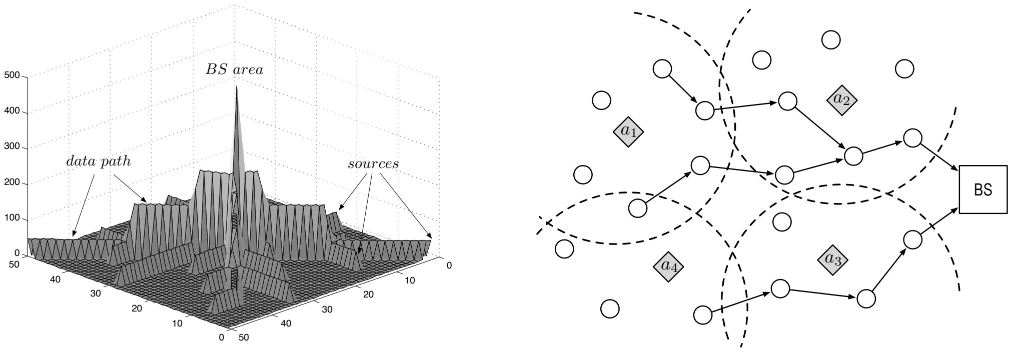
[9] Most sensor networks only monitor a particular type of event. In a multi-event sensor network, if the adversary wants to discern between different types of events, he might turn to the analysis of other features like those presented in Section 1.3.2.

use of hashing functions to obfuscate the addressing fields in the packet headers. However, it was only later that they realised that this type of countermeasure was insufficient protection from adversaries performing both content analysis as well as more sophisticated traffic analysis attacks.

Local adversaries are interested in finding the base station and thus traceback attacks are no longer useful. Instead, the adversary must determine the direction of the communications flow. To that end, he might first turn to time correlation attacks, where the idea is to determine the next node in the communication path by observing the time difference between the transmission of a node and its neighbours. Based on the assumption that a node transmits a message immediately after it is received, the attacker can determine the next node in the path by observing the neighbouring node which transmitted in the shortest space of time. Since event data is always addressed to the base station, by moving to the forwarding node, the attacker is capable of reducing the distance to the sink. This process is then repeated at each intermediate node until finally the attacker finds his target. Additionally, the adversary can opt to use a rate monitoring attack to reach the base station. This attack is based on the fact that the transmission rate in the vicinity of the base station is higher than in remote areas because of the use of multi-hop communications (see Figure 1.7a). A sensor node close to the sink must serve as an intermediary for remote nodes, thus increasing the number of packets it transmits. Consequently, before making a decision on his next move, the adversary observes the number of transmissions of a node and its neighbours. After a sufficient number of observations the attacker can deduce the neighbour which is most likely to be closer to the base station and move accordingly.

Similarly, a global adversary uses rate monitoring attacks to infer the location of the base station. The use of an adversarial network allows him to compare the number of packets observed in each area without having to move around in the field. The adversarial nodes recording a higher number of packets reveal to the attacker which areas are close to the base station. In Figure 1.7b we illustrate the deployment of an adversarial network $\{a_1, \ldots, a_4\}$ observing the communications within its hearing range, which is represented by dashed semi-circles. As the adversarial nodes a_2 and a_3 overhear a higher number of transmissions[10], they are more likely to be close to the base station.

[10] The use data-aggregation algorithms may reduce the number of transmissions in the vicinity of the base station but the traffic pattern would still be pronounced in the presence of numerous data sources.



(a) Number of transmissions in a WSN implementing single-path routing with 15 data sources and a single base station

(b) Adversarial network (a_i) deployed to monitor the transmissions of the legitimate network

Figure 1.7: Traffic Rate Monitoring in a Typical WSN

Finally, note that we are addressing homogenous sensor networks, where all the sensor nodes have the same role, i.e., sensing, reporting, and relaying data. However, in heterogeneous sensor networks the communication pattern may differ slightly depending on the configuration of the network. For example, in a hierarchical configuration, sensor nodes are organised into clusters controlled by a cluster head which makes all organisational decisions, like routing the data sensed by the cluster members to the base station. Therefore, the adversary might be interested in finding nodes with a particular functionality, like the cluster head. This thesis concentrates on the first type of network configuration although some of the approaches and solutions that have been developed may also be applicable to the second type of network.

1.5 Goals and Organisation

Wireless sensor networks bring tremendous benefits to our society due to their ability to link the virtual world and the real world. Unfortunately, the deployment of such context-aware technologies may also involve a number of risks and threats that need to be carefully assessed before they are socially accepted. A major impediment to social acceptance is the potential risk of privacy violations that wireless sensor networks entail. This is precisely the main focus of this thesis and our research efforts are aimed at *facilitating a privacy-aware integration of wireless sensor networks* in our daily lives.

In order to ease the acceptance of wireless sensor networks it is important to build trust in the technology and the underlying mechanisms used to enforce privacy. Therefore, we deemed it necessary to analyse whether it is strictly necessary to devise new solutions to the location privacy problem in WSNs, especially when there is an extensive body of research into anonymous communications systems which are capable of providing a solid privacy protection to their users. The adaptation of these solutions would imply both strong privacy and users' trust. Consequently, one of the goals of this thesis is to *study the suitability and applicability of computer-based anonymous communication systems* to the source- and receiver-location privacy problems in wireless sensor networks.

Another goal of this thesis is to *analyse and categorise the existing location privacy solutions in WSNs*. This is important to gain insight into the techniques used to counter the various types of adversaries as well as to identify the advantages, limitations and open problems of the current state-of-the-art. In the case that adaptation of computer-based anonymity solutions is infeasible or impractical for some reason, the analysis undertaken will help to devise improved solutions for protecting both the location of the data sources and the base station.

Finally, we need to *develop novel source- and receiver-location privacy solutions and evaluate them*. The evaluation of the devised solutions must include not only the level of protection they are capable of achieving but also how much their application affects the performance of the network. Since these networks are highly resource-constrained, it is strictly necessary to provide mechanisms to balance between the level of protection and the usability and survivability of the network as it is useless to provide perfect privacy if the network functionality is severely impaired.

Regarding the applicability of the results arising from this thesis, we concentrate on scenarios where the network consists of a large number of static nodes and there is a single base station. Although this is the most typical configuration at the time of writing, some of the results may also be valid for different network configurations with several base stations and a few mobile nodes. With respect to the support of hardware platforms, we consider that our solutions are suitable for the typical and high-performance configurations available today (see Section 1.1) but we believe some of the solutions can even fit into the extremely constrained class of sensor nodes since they are based on lightweight computations and demand little memory space. Basically, the only requirement is that nodes are capable of performing hop re-encryption. Nevertheless, battery consumption is usually a limitation for almost any sensor platform running location privacy solutions.

1.5.1 Main Contributions

Next we present a list of the major contributions of this thesis to the area of location privacy in wireless sensor networks:

- Highlight the importance of the contextual privacy problems affecting WSNs due to their particular mode of operation with a particular focus on the location privacy of the data sources and the base station.

- Study of the suitability of existing computer-based anonymous communication systems to the location privacy problem in WSNs, paying special attention to the particular requirements, limitations and adversarial models considered in both scenarios.

- Analysis of the current state-of-the-art in location privacy solutions for sensor networks specially focusing on the advantages and limitations of the solutions, and the identification of open problems and research gaps.

- Definition of an exhaustive taxonomy of location privacy solutions in WSNs based on the property the solution is aiming to protect, the capabilities of the adversarial model, and the main features of the solution.

- Development and evaluation of a context-aware mechanism which can be integrated with existing routing protocols to enhance source-location privacy against local adversaries with minimal impact on network performance.

- Development and evaluation of a receiver-location privacy mechanism based on traffic normalisation and routing table obfuscation that is capable of offering protection, for the first time, against both local eavesdroppers and active attackers capable of retrieving the routing tables of a portion of the nodes in the network.

1.5.2 Thesis Outline

In this first chapter, we have introduced the concept of Wireless Sensor Networks by illustrating their specific features, including hardware limitations, communication model and routing protocols. We have also provided a quick overview of security threats and countermeasures in this paradigm. Moreover, we have introduced the two main privacy research areas in wireless sensor networks and next we have focused on context-oriented privacy and more precisely on the location privacy problem, which have been sufficiently explained and motivated. Finally, we have presented the goals and contributions of this thesis.

Before devising new tailored solutions to a given problem it is necessary to analyse whether these new solutions are strictly necessary. This is especially important when there is a well-founded area with a number of solutions to problems which are closely related to the one being tackled. This is precisely the motivation of Chapter 2. Since location privacy problems in WSNs are caused by their particular communication pattern, these problems may be countered by traditional traffic analysis protection mechanisms devised for computer networks. This hypothesis has been rejected by several authors by simply claiming that sensor nodes cannot withstand the heavy computational overhead imposed by these solutions. However, this reason alone is insufficient to exclude them from the WSN domain as new sensor nodes with more capacity can be built. Therefore, this chapter studies which anonymity properties are most suitable to fit the particular features and requirements of location privacy in WSN and, on top of that, it analyses some well-known computer-based anonymity solutions in order to give insight into their overhead and possible limitations to the application of the network.

Chapter 3 provides a literature review and analysis of the existing solutions for location privacy in WSNs. The presentation of this chapter is guided by several criteria that allows us to classify solutions according to the assets that demand protection, the capabilities of the adversary, and their most distinguishing features. First, we analyse a set of solutions that have been devised to protect the identity of the nodes during data transmission. Next, we concentrate on solutions aimed at hiding the location of the data sources and the base station by changing the normal communication pattern of the network. These solutions are further divided depending on the capabilities of the adversary under consideration: local or global eavesdropper, and internal adversaries. For each individual solution we present some advantages and limitations. This has helped to identify pitfalls, open problems, and possible lines for pushing forward the state of the art. As a result we present a complete taxonomy of solutions and discuss some possible lines of actuation which are exploited in the following chapters.

The Context-Aware Location Privacy (CALP) is presented in Chapter 4. This mechanism benefits from the intrinsic nature of sensor nodes of being able to feel their environment to detect the presence of a mobile adversary in the network deployment area. The idea is to anticipate the movements of the adversary and modify the routing paths in order to minimise the number of packets he is able to capture. The scheme has been successfully applied to protect source-location privacy in the presence of local adversaries with different moving strategies. In particular, we have developed two versions of the CALP scheme, which differ on the penalty imposed on paths traversing the area where the adversary is located. Since the proposed scheme is only triggered in the presence of the adversary, it considerably reduces the overhead imposed on the network compared to previous solutions.

Chapter 5 describes a receiver-location privacy solution called the Homogeneous Injection for Sink Privacy with Node Compromise protection (HISP-NC) scheme. The proposed solution consists of two complementary schemes that deal with adversaries capable of observing the communications in a limited area of the network as well as inspecting the routing tables of a portion of the nodes. The first scheme injects controlled amounts of fake traffic to probabilistically hide the flow of real messages towards the base station. This scheme on its own provides an adequate protection level against local eavesdroppers but is useless if the adversary is is capable of gaining the information contained in the routing tables of a few nodes. The second scheme provides, for the first time, some means of protection against this type of threat by perturbing the routing tables of the nodes in such a way that inspection attacks are not trivial but real messages reach the base station within a reasonable time frame.

Finally, Chapter 6 summarises the contributions of this thesis and presents some potential lines of future work as well as some open research problems that demand further attention from the research community.

1.6 Publications and Funding

The main contributions of this dissertation have been published in various journals and conferences, both national and international. Next, we provide a list of the main contributions organised by the type of publication:

Journal articles ISI-JCR

- Ruben Rios, Jorge Cuellar, and Javier Lopez. Probabilistic receiver-location privacy protection in wireless sensor networks. *Information Sciences*, Accepted for publication. Impact Factor: 3.89

- Ruben Rios and Javier Lopez. Exploiting Context-Awareness to Enhance Source-Location Privacy in Wireless Sensor Networks. *The Computer Journal*, 54(10):1603–1615, 2011. doi: 10.1093/comjnl/BXR055. Impact Factor: 0.79

- Ruben Rios and Javier Lopez. (Un)Suitability of Anonymous Communication Systems to WSN. *IEEE Systems Journal*, 7(2):298 – 310, June 2013. ISSN 1932-8184. doi: 10.1109/JSYST.2012.2221956. Impact Factor: 1.27

- Ruben Rios and Javier Lopez. Analysis of location privacy solutions in wireless sensor networks. *IET Communications*, 5:2518 – 2532, 2011. ISSN 1751-8628. doi: 10.1049/iet-com.2010.0825. Impact Factor: 0.83

International conference papers

- Javier Lopez, Ruben Rios, and Jorge Cuellar. Preserving receiver-location privacy in wireless sensor networks. In Xinyi Huang and Jianying Zhou, editors, *Information Security Practice and Experience (ISPEC 2014)*, volume 8434 of *LNCS*, pages 15–27, Fuzhou (China), May 2014. Springer, Springer. doi: 10.1007/978-3-319-06320-1_3

- Ruben Rios, Jorge Cuellar, and Javier Lopez. Robust Probabilistic Fake Packet Injection for Receiver-Location Privacy in WSN. In S. Foresti, M. Yung, and F. Martinelli, editors, *17th European Symposium on Research in Computer Security (ESORICS 2012)*, volume 7459 of *LNCS*, pages 163–180, Pisa (Italy), Sept. 2012. Springer. doi: 10.1007/978-3-642-33167-1_10

- Ruben Rios and Javier Lopez. Source Location Privacy Considerations in Wireless Sensor Networks. In Lidia Fuentes, Nadia Gámez, and José Bravo, editors, *4th International Symposium of Ubiquitous Computing and Ambient Intelligence (UCAmI'10)*, pages 29 – 38, Valencia (Spain), Sept. 2010. ISBN 978-84-92812-61-5

Book chapters

- Ruben Rios, Javier Lopez, and Jorge Cuellar. Location Privacy in WSNs: Solutions, Challenges, and Future Trends. In *Foundations of Security Analysis and Design (FOSAD) VII*, volume 8604, pages 244–282. Springer, 2014. doi: 10.1007/978-3-319-10082-1_9

National conference papers

- Ruben Rios, Jorge Cuellar, and Javier Lopez. Ocultación de la estación base en redes inalámbricas de sensores. In Jesús E. Díaz Verdejo, Jorge Navarro Ortiz, and Juan J. Ramos Muñoz, editors, *XI Jornadas de Ingeniería Telemática (JITEL 2013)*, pages 481–486, Granada, Oct 2013 2013. Asociación Telemática. ISBN 978-84-616-5597-7

- Ruben Rios and Javier. Lopez. Adecuación de soluciones de anonimato al problema de la privacidad de localización en WSNs. In R. Uribeetxeberria U. Zurutuza and I. Arenaza-Nuño, editors, *XII Reunión Española sobre Criptología y Seguridad de la Información (RECSI 2012)*, pages 309–314, San Sebastian (Spain), Sept. 2012

Additionally, this thesis has motivated the development of a WSN simulator in MATLAB to support the analysis of the proposed solutions in terms of usability and privacy protection level. This tool consists of a discrete-event simulation environment for different network configurations (i.e., topology, connectivity, etc.) and a variety of attacker strategies (i.e., random, traceback, etc.). The simulator enables multiple data sources simultaneously together with the presence of various attackers moving in the field. An important feature of this simulator is that it obviates the lower levels of the communications stack and concentrates on the routing layer since our goal is to demonstrate the feasibility of the models and protocols that have been devised.

Finally, some other papers have been written in parallel with the main contributions of the dissertation. These papers are not directly related to the topic of the thesis but are related to privacy and information leakage in various environments. The contributions are sorted by date and they include both journal and conference papers:

- Isaac Agudo, Ruben Rios, and Javier Lopez. A Privacy-Aware Continuous Authentication Scheme for Proximity-Based Access Control. *Computers & Security*, 39, Part B:117–126, November 2013. ISSN 0167-4048. doi: 10.1016/j.cose.2013.05.004

- Ruben Rios, Jose A. Onieva, and Javier Lopez. Covert Communications through Network Configuration Messages. *Computers & Security*, 39, Part A:34–46, 2013. ISSN 0167-4048. doi: 10.1016/j.cose.2013.03.004

- Jorge Cuéllar, Martín Ochoa, and Ruben Rios. Indistinguishable Regions in Geographic Privacy. In *Proceedings of the 2012 ACM Symposium on Applied Computing (SAC'12)*, pages 1463–1469, Riva del Garda (Trento), Italy, March 2012. ACM. ISBN 978-1-4503-0857-1. doi: 10.1145/2245276.2232010

- Ruben Rios, Jose A Onieva, and Javier Lopez. HIDE_DHCP: Covert Communications Through Network Configuration Messages. In *Proceedings of the 27th IFIP TC 11 International Information Security Conference*, volume 376 of *IFIP AICT*, pages 162–173. Springer, Heraklion, Crete (Greece), June 2012. doi: 10.1007/978-3-642-30436-1_14

- Ruben Rios, Isaac Agudo, and Jose L. Gonzalez. Implementación de un esquema de localización privada y segura para interiores. In Yannis Dimitriadis and María Jesús Verdú Pérez, editors, *IX Jornadas de Ingeniería Telemática (JITEL'10)*, pages 237 – 244, Valladolid (Spain), Sept. 2010. ISBN 978-84-693-5398-1

This dissertation has been possible thanks to the support and funding of the Spanish Ministry of Education through the National Programme for Training Human Resources under the FPU (Training of University Lecturers) fellowship. Moreover, some parts of this work have been supported by and applied to different National and European projects: ARES (CSD2007-00004), SPRINT (TIN2009-09237), IOT-SEC (ACI2009-0949), and NESSoS (FP7 256890).

Chapter 2

Suitability of Computer-based Anonymity Systems

This chapter analyses the suitability of anonymous communication systems for the protection of location privacy in WSNs. Before devising new solutions we deem it necessary to understand whether existing solutions are able to provide an adequate protection level in the sensors' domain. Several authors [94, 140] have established that such systems are not applicable to the sensor domain with vague arguments about the prohibitive hardware requirements of anonymous communication systems. Notwithstanding, given the extensive literature on anonymous communication systems and the maturity of research in the field, we believe that excluding the solid protection mechanisms provided by these systems without proper analysis would be a serious mistake, especially considering that the capabilities of sensor nodes can improve considerably in the future.

To decide whether or not anonymous communication systems are truly unsuitable for WSNs it is necessary to strictly analyse the requirements, goals, and techniques proposed by these systems, as well as the particular features and requirements imposed by the new scenario. First, we study which anonymity properties better suit the location privacy problem in WSNs given the capabilities and strategies of the adversary. Next, we examine both centralised and decentralised anonymous communications systems in order to determine their limitations and imposed overhead. Finally, we briefly discuss the factors that may limit the application of the analysed solutions in the realm of sensor networks.

2.1 Anonymous Communication Systems

Data communication networks allow us to establish online transactions with remote entities. These networks rely on a series of protocols that use addressing information to identify the parties which are intended to receive or route messages on their way to the other communication end. Even when application-layer data are properly secured using end-to-end encryption, the addressing information (e.g., IP and MAC addresses) is sent in clear text in order to enable data routing at intermediate nodes. These addresses seldom change and appear in every single packet, which allows anyone observing the communication to correlate all the transactions belonging to a user. Moreover, some addresses are unique to a specific device, which can be ultimately linked to a particular individual thus severely compromising his/her privacy. Additionally, an ambitious adversary can perform more sophisticated traffic analysis attacks, such as monitoring the volume of packets being sent or received, in order to obtain more detailed information about the user.

Anonymous communication systems were devised precisely to protect users' privacy in the presence of highly motivated adversaries performing traffic analysis attacks. These systems are based on a number of techniques and mechanisms, usually built on top of cryptographic primitives, which are intended to conceal the addresses of the users as well as any other information associated with their identity that can be extracted by observing of the traffic generated from their interactions with other users or systems.

Usually, anonymous communication systems consider a scenario like the one depicted in Figure 2.1. In this setting, a set of senders use a data communication network to send messages to a group of recipients and the goal is to ensure that an attacker cannot retrieve sensitive information about the communicating parties from the observation of a portion of the network or the system as a whole. The attacker may be interested in determining different types of information from the set of potential senders and recipients and the system struggles to offer some anonymity-related properties that are intended to prevent that disclosure of information.

Pfitzmann and Hansen [102] provide a comprehensive and widely-accepted terminology for describing privacy-related concepts. We adopt and review these definitions for our analysis, as having a complete understanding of anonymity properties is essential for a detailed analysis of any anonymity solution.

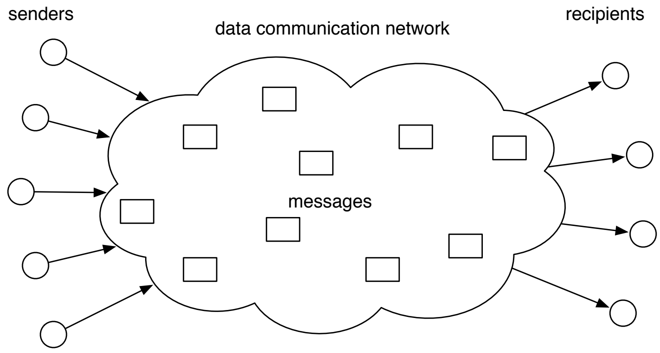


Figure 2.1: Communications setting

2.1.1 Anonymity Terminology

Defining privacy is usually difficult because of the subjectivity of the term. This concept has different interpretations and nuances depending on many different factors such as socio-economical condition, level of educational, religious beliefs, and so on. A simple yet renowned definition considers privacy as the right to be left alone [143], however, new definitions have appeared as new ways of invading privacy have emerged. In the area of information technology, privacy can be defined as the right of individuals (or entities) to control the disclosure, processing, and dissemination of information about themselves. Therefore, privacy is closely related to anonymity because it describes the desire of an individual to remain unidentified when performing some action. However, anonymity is not the only useful property to accomplish privacy. Next we review the most relevant privacy-related properties based on the terminology from Pfitzmann and Hansen [102].

Anonymity can be defined as the state of being not sufficiently identifiable within a set of subjects (i.e., the anonymity set) with potentially the same attributes as the original subject. In other words, anonymity mechanisms prevent the disclosure of the identity of the individual who performed a particular action (i.e., the attribute) by having a set of potential actors. Clearly, if all the members in the anonymity set are equally likely to be the author of the action, the anonymity becomes stronger as the size of the anonymity set grows. Ideally, the probability that an adversary can successfully determine the actual entity who performed the action is one over the size of the anonymity set. However, in practice, not all members in the anonymity set are equally likely to be the actual author. In the landscape of anonymous communication systems, the action usually refers to the transmission or reception of messages. Therefore, a sender may be anonymous within a set of potential senders and, similarly, a recipient may be anonymous within a set of potential recipients. These properties are known as sender and receiver anonymity, respectively.

Another important property for the protection of individual privacy is *unlinkability*. Unlinkability of two (or more) items of interest means that an adversary cannot sufficiently distinguish whether these items are related or not. By definition, the items of interest may be any element of the system, such as entities or messages. Therefore, we may encounter different types of unlinkability. Commonly, anonymous communication systems strive for the unlinkability of the sender and the receiver[1], which provides the communicating parties with the ability to hide with whom they communicate. This is usually known as relationship unlinkability and it is useful in the presence of external observers trying to infer information about the preferences of an individual. When a user accesses an online service (e.g., websites) regularly this reveals information about his or her interests. For example, daily visits to a particular online newspaper might reveal a right- or left-wing ideology. Besides, relationship unlinkability suggests that even when the sender and the receiver can each be identified as participants in a communication, they cannot be recognised as communicating with each other. This implies that the unlinkability property is stronger than anonymity.

Finally, undetectability and unobservability are properties that aim to protect the items of interest themselves. *Undetectability* of an item of interest means that the attacker cannot sufficiently determine whether a particular item exists or not. Similarly, *unobservability* means undetectability of the item against all external entities and, additionally, anonymity even against other subjects involved in the item of interest. In anonymous communication systems, the cited properties usually refer to messages as the objects of interest. Therefore, undetectability aims to prevent an attacker from determining whether (real) messages are being transmitted. On the other hand, unobservability not only implies that an external attacker cannot detect the presence of messages but also that other senders/receivers cannot sufficiently determine who is sending/receiving the messages. A sender is unobservable when the attacker is not able to determine

[1] The attacker model determines the sort of unlinkability required. For example, message unlinkability is important for preventing a server from linking multiple requests from the same source in order to avoid user profiling.

whether any of the senders is transmitting real messages. Likewise, the recipient is unobservable if the adversary cannot conclude whether it is receiving real data messages.

2.1.2 Anonymity Properties in WSNs

Prior to the analysis of traditional anonymous communication systems, here we discuss the need for and suitability of the anonymity properties described in Section 2.1.1 with reference to the location privacy problem in WSNs. Among the various pieces of sensitive information that might be gathered by an observer of the communications, we concentrate on the location of the nodes reporting or receiving event messages since their location can be determined by means of traffic analysis.

Since the main focus of anonymous communication systems is hindering traffic analysis, in principle, these systems might also be ideal for protecting the location of the data sources and the base station in WSNs. However, there are several limitations to the application of traditional solutions in the sensors domain. Here we concentrate on discussing which of the design principles that have guided the development of traditional anonymity systems are meaningful for the protection of location privacy in sensor networks.

Firstly, anonymity is only necessary in certain circumstances in WSNs, even being detrimental to the correct operation of the network in some cases. Source anonymity with respect to the recipient is not beneficial for the operation of the network because in most application scenarios the base station (i.e., the recipient) needs to be aware of the original data sender. The base station uses the source ID for the management and control of the environment being monitored. Without this information, the base station cannot identify the original source of the data and thus it is unable to provide the network administrator with relevant information about the sensor field. Notwithstanding, sender anonymity might be useful to prevent external observers from determining the data source. Hiding this information helps in the protection of the location of events against adversaries who have access to a map of the network or who have created one by patently eavesdropping on every single network node. This problem can be prevented by occasionally changing the nodes' IDs for a pseudonym[2] in such a way that even if the attacker obtains such a map, it is rendered useless when the

[2] A pseudonym is an identifier used instead of the original ID.

current identifiers change their values. There are already several approaches that consider the use of dynamically changing pseudonyms for WSNs [92, 99]. In these solutions, the pseudonyms are known to the base station and it is, therefore, able to spot the occurrence of events in the field. Finally, it might be useful to prevent compromised sensor nodes (i.e., nodes controlled by the attacker) to gain access to the real source ID. Since remote sensor nodes rely on intermediate nodes to forward their data, if any of these en-route nodes are compromised they might get access to the source ID. This problem has also been considered by some authors [103]. Therefore, source anonymity is only necessary in certain circumstances in WSNs.

Moreover, given the existing communications model in sensor networks, the sender-receiver unlinkability property does not make much sense. The normal operation of the network implies many-to-one communications, where any sensor node is a potential sender and the base station is the only receiver. Therefore, the property of relationship unlinkability is lost because, in any event data transmission, the base station is one of the participants. In traditional anonymous communication systems, relationship unlinkability is important in terms of the identity of the sender and the recipient because it gives away information about the behaviour and preferences of users. Contrarily, in the case of location privacy in WSNs, all sensor nodes transmit to a single base station and therefore there is no such information gain. Here, the important issue is to determine the location of these nodes and this cannot be done by simply analysing packets in transit unless this information is given either in the headers or the payload. However, the attacker is assumed not to have access to the payload because it is cryptographically protected, but the header might provide some information on the source. This issue becomes problematic only in the case that the adversary already knows the network topology but, as aforementioned, this problem is related to source anonymity, not to unlinkability. Similarly, in the case of the receiver, since we are focusing on flat and homogeneous sensor networks with a single base station, which is in charge of collecting all the data, there is actually no need to indicate in the packets which node is the final recipient of the data. Finally, message unlinkability is also unnecessary and counterproductive for the same reasons as source anonymity.

In fact, the most natural property for the protection of location privacy is unobservability rather than unlinkability. By hiding the presence of the nodes reporting event data or receiving it, we can prevent the attacker from determining the location of events in the field and the location of the base station. More precisely, the attacker will be unable to obtain the location of the communicating nodes if he is unable to sufficiently detect the presence of event messages in the network. Clearly, if the attacker is not able to ascertain the existence of messages containing event data, he will not be able to determine which node is the sender or the recipient of that message under the assumption that he has no information other than the observed traffic. Note that the attacker could benefit from other sources of information, such as visual recognition of the event or previous knowledge about the nature of events being monitored, to aid him in the search. However, this is beyond the scope of this thesis and we assume that the adversary has no prior knowledge about the deployment of the network or visual information about the events taking place in the field.

In summary, we can state that some anonymity properties are not suitable or necessary for the protection of location privacy in WSNs. Notwithstanding, the following sections will delve into each specific solution regardless of their main design goal in order to have a clearer understanding of the particular features, the imposed overhead, and the techniques proposed by renowned anonymous communication systems originally devised for the Internet. By doing this we will finally be in a position to assess the real limitations or potential applicability of these systems to preserve location privacy in the sensors' domain.

2.1.3 Classification of Solutions

Many outstanding anonymous communication systems have been devised to hinder traffic analysis and thus improve the privacy protection in online communications. These systems have been designed with different goals in mind and, so, they pursue different anonymity properties. We propose a taxonomy of solutions which takes into consideration three major features, namely: (1) the main desired goal in the design of the anonymous communication system, (2) the architectural design, and (3) the principal techniques used to reach the goals. This taxonomy is presented in Table 2.1 but, for the sake of simplicity, only the most commonly used techniques have been represented.

Among the multitude of anonymous communication system designs, we have selected several outstanding solutions that introduce various distinguishing features and countermeasures that are addressed for different adversarial models.

	Main goal	**Architecture**	**Techniques**							
			SK	PK	LE	RN	PD	PR	FT	MB
Single-proxy [10]	Sender Anonymity	Centralised	√	-	-	√	-	-	-	-
Mixes [26]	Unlinkability		-	√	√	-	√	-	-	-
Onion routing [108]			√	√	√	-	-	-	√	-
Tor [43]			√	√	√	-	-	-	-	-
Crowds [109]	Sender Anonymity	Decentralised	√	-	-	√	-	-	-	-
Hordes [70]			√	√	-	√	-	-	-	√
GAP [14]	Unobservability		√	√	-	√	√	√	√	-
DC-nets [27]			√	-	-	-	-	-	-	√
Herbivore [48]			√	√	-	-	-	-	-	√

Notation	
SK	Symmetric-key encryption/decryption
PK	Public-key encryption/decryption
LE	Layered encryption
RN	Source identity renaming
PD	Temporal packet delay
PR	Packet replay
FT	Fake traffic injection
MB	Multicast or broadcast communications

Table 2.1: Classification of Anonymous Communication Systems

From an architectural point of view, these solutions can be categorised as either centralised or decentralised. Centralised solutions are those in which the communicating parties are not an active part of the anonymity system, namely, there is a set of devices in between senders and recipients which are responsible for forwarding and anonymising the communications. Contrarily, in decentralised solutions each user collaborates in the forwarding process to conceal his own communications and the communications of other participants. Some of the analysed solutions are partially decentralised because they rely on a central server which is in charge of providing all the information necessary to communicate with other members of the system or external entities, while other solutions are fully decentralised and not dependent on a central authority. Also, in these solutions, data recipients might be part of the anonymous communication network or external entities.

The proposed categorisation also takes into consideration the main goals pursued by these solutions. It is worth mentioning that some of these solutions might have been designed with several goals in mind but only the most relevant ones are included in the table for the sake of simplicity. For example, mix-net approaches aim to provide sender-receiver unlinkability although they might also ensure sender anonymity. Note that when we refer to sender anonymity, we usually refer to anonymity with respect to the data recipient. In many situations a client is willing to gain access to a particular service but is reluctant to provide his/her real identity to a potentially untrustworthy service provider because of the concern of being tracked or profiled for illegitimate purposes, such as price discrimination.

Finally, note that the various techniques employed by these solutions could be used to further break down into new categories. For example, the presented solutions could be organised into high-latency or low-latency solutions depending on whether the systems introduce large delays before relaying the packets received. Instead, to guide the exposition and analysis of the solutions in the following sections, we concentrate on the architectural (i.e., centralised or decentralised) perspective, which is more natural and consistent with the evolution of research in the field of anonymous communication systems. Nonetheless, the notation presented for the various techniques will be used during the overhead analysis of the systems.

2.2 Centralised Schemes

Centralised anonymous communication systems rely on a set of partially-trusted[3] devices which are responsible for conveying data from senders to receivers in a privacy-preserving manner. Whenever a user wants to send data anonymously to another party, it does so by contacting anyone of the devices comprising the anonymity network. The anonymity relay(s) will eventually deliver the received data to the final destination on behalf of the user, thereby obscuring the actual data source. In short, the communication ends (i.e., sender and recipient) are not members but clients of centralised anonymity systems.

As a result, an attacker capable of observing all the nodes involved in the system can easily spot the communication ends. This is the case of global adversaries, who can observe all the transmissions in the network. Consequently, centralised anonymous communication systems are unable to protect themselves from such a powerful adversarial model. Thus, only local observers and internal attackers will be considered in this section.

[3] Honest relays may be forced to reveal information under legal compulsion.

2.2.1 Single-proxy

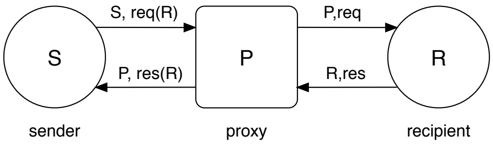
Several single-proxy solutions (e.g., Anonymizer [10]) have been proposed to allow Internet users to access online services, like surfing the web, without disclosing their identity to service providers[4]. In other words, single-proxy solutions aim to provide source anonymity against potentially dishonest service providers mostly interested in tracking and profiling.

These solutions are based on a trusted third party which acts as an intermediary between the user and the real destination. The operation is very simple as depicted in Figure 2.2a. Whenever a user (i.e., the sender) wants to communicate with a server (i.e., the recipient) it issues a message to the third party (i.e., the proxy) informing them about the intended recipient and the original request. Then, the proxy forwards the user request to the server but first, removes the true source of the request. In this way, as far as the recipient is concerned the proxy appears to be the original data sender, thus hiding the true identity of the true sender to the destination. Finally, the recipient responds to the message as if it came from the proxy, which needs to keep track of connections to send the reply back to the user. Additionally, some single-proxy solutions create an encrypted tunnel from the user to the proxy[5] in order to prevent eavesdropping on that link.

From a computational point of view single-proxy solutions introduce relatively low overhead, which is an interesting feature for hardware-constrained sensor nodes. Source sensor nodes are only required to transmit their data to a proxy node and, in the worst case, encrypt it. A proxy node, on the other hand, must decrypt the data from different sources (if encrypted) and change the source ID of incoming packets with their own identifier. This process is referred to as renaming and must be done for every single message. Note that, given that the communication is assumed be unidirectional from sensor nodes to the base station only, a proxy node does not need to keep track of messages in order to send a reply back to the data source. Additionally, the data source could opt to apply end-to-end encryption in order to protect the data on its way from the proxy to the base station. This would imply a extra encryption operation on both ends of the communication but could also relief the proxy from decrypting in the first

[4] In most cases, when we talk about identity we are actually referring to any information that is linked to or can be used to identify an individual, usually an IP address.

[5] The communication from the proxy to the server can also be encrypted if the server provides that functionality (e.g., using HTTPS).



(a) Single-proxy Communication

Node	**Case**	
	Best	Worst
Sources	–	$2SK$
Proxies	$1RN$	$1RN + 1SK$
Sink	–	$1SK$

(b) Single-proxy Overhead

Figure 2.2: Single-proxy Solutions

place. The table in Figure 2.2b summarises the total number of operations that different network nodes will need to perform depending on the (best or worst case) scenario. Note that the values in the table refer to a single message transmission. The terminology used in this and subsequent tables is consistent to that described in Table 2.1.

Despite the low overhead introduced, given the threat model under consideration, a single-proxy approach alone cannot prevent the location privacy problem in WSNs. In the case of a local observer, this solution is unsuitable because the adversary uses strategies that lead him to the target regardless of the packet contents or headers. When looking for a source node, the attacker uses a traceback strategy based on the angle of arrival of signals. Whereas an adversary willing to find the sink, can simply turn to rate-monitoring or time-correlation attacks. To prevent these attacks, it is necessary to prevent the use of persistent paths by randomising the routes to the proxy, using different proxies for each new message, etc. In fact, Phantom Routing [60] and other solutions use random intermediate nodes (similar to proxies) from where the source data is finally routed to the base station. However, the protection mechanism does not reside in renaming or data encryption but on the selection of random intermediate nodes which lead to ephemeral routes that confuse the adversary.

On the other hand, the use of renaming and end-to-end encryption provides some protection against internal adversaries. However, the level of protection is insufficient considering that only the nodes located after the proxy node would not have access to the true source identifier. Also, note that the attacker may be able to identify a proxy node and compromise it, thus easily gathering the identifiers of all the source nodes using the proxy. A potential countermeasure for this is to have proxies at various distances from the base station forming a multi-tier proxy architecture but this is something that will be discussed in the next section.

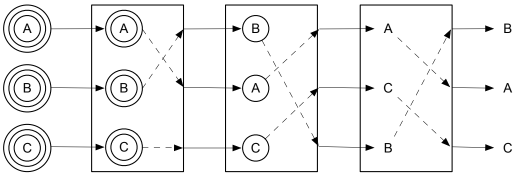
2.2.2 Mixes

A mix is a store-and-forward device that receives public-key encrypted messages and after a sufficiently long period of time has passed, it outputs a re-ordered batch of all messages. In this way, mixes hide the correspondence between input and output messages because of temporal storage and decryption. This type of high-latency anonymity solutions were originally devised by D. Chaum [26] for non-interactive online communications, such as anonymous e-mail transmissions. Usually, mixes are arranged and selected in series (i.e., mix cascade) or deployed as a fully connected network and picked in a random order (i.e., mix network). In such arrangements a single honest mix preserves the unlinkability between inputs and outputs along the whole path.

A mix cascade is depicted in Figure 2.3a, where messages A, B, and C are encrypted with the public keys of the mixes they will traverse in reverse order. Each mix removes its corresponding encryption layer and outputs a lexicographically ordered batch of messages after a sufficiently large amount of time to prevent the correlation of inputs and outputs. After decryption, each mix adds a block of random bits at the end of each message to maintain their size constant. Additionally, both users and the mixes themselves can introduce dummy messages to hide the number of messages sent and received at each point.

The implementation of a mix-based solution over WSNs presents several limitations in terms of the computational overhead introduced. Source nodes are required to perform N public-key operations per transmitted packet[6]. Additionally, source nodes must have a global knowledge of the network topology in order to be able to determine the transmission paths and perform the public-key encryptions in the right order. Moreover, this implies that source nodes must store the public keys (P_K) of all potential mix nodes. Clearly, the size of the mix network has a tremendous impact on the number of operations and the amount of memory needed to store all the information required to satisfy the mix-based model. However, the recipient of messages, which is the most powerful device of the network, is freed from performing any operation.

[6] An extra operation, being it symmetric or asymmetric, satisfies the end-to-end encryption principle.



(a) Communications in a Mix Cascade

Node	**Requirements**	
	CPU	RAM
Sources	$N \cdot PK$	$M \cdot P_K + topo$
Mixes	$1PK + 1PAD$ $+1PERM$	$1S_K + T \cdot mess$

(b) Mixes Overhead

Figure 2.3: Mix networks

Moreover, each intermediate node has to perform 1 public-key decryption per received packet as well as temporarily store a number of messages (T) that depends on the number of events in the field. Also, the closer the nodes are to the base station the higher the traffic rate is. But, the amount of memory fitted inside a typical sensor node is insufficient to accommodate a large number of messages. Besides, to preserve message indistinguishability and prevent leaking the direction of message flows, packets are padded (PAD) after decryption. This not only requires more computation but also more wirelessly transmitted bits. Finally, many WSN applications require real-time monitoring capabilities but mixes introduce significant delays at each node thus precluding their use in these scenarios.

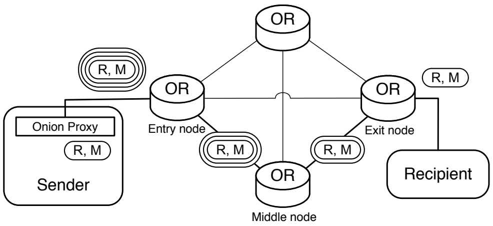
A summary of results is provided in Table 2.3b, where, for the sake of simplicity, only the best case scenario is represented. Note that the worst case (i.e., source nodes performing end-to-end encryption) implies that, for every message transmitted, a source node performs an extra cryptographic operation and, moreover, the base station must share keys with all potential source nodes. Furthermore, we do not consider scenarios where the destination responds to the source. In such cases, the base station would perform the same number of operations as a source node and mix nodes would have to perform roughly the same number of operations as in the forward path. Additional terminology appears in this table: S_K is the node's own private key, N is the number of nodes in a path, and M is the number of mixes. Additionally, *topo* refers to the topology of the network and $T \cdot mess$ indicates the temporal storage of messages. We acknowledge that some of these values may vary over time depending on the workload of the network. However, our goal is not to make an exhaustive and accurate to the milliwatt overhead study as this would require having real implementations running on the motes. Rather we are more interested in gaining an overall idea on the potential cost of deploying these solutions.

In addition to the high computational and memory overhead imposed by mix-based schemes, there are other limitations that hinder the successful deployment of the mixes, taking into account the types of adversaries considered in WSNs. The main aspect is that the adversary wins if he is able to obtain the location of either the source node or the base station, contrarily to the goal of the adversary in the traditional scenario, where he wants to determine whether a particular sender is communicating with a particular recipient. In such scenarios the temporal mix of messages provides the desired property but, in sensor networks, it makes no difference whether the adversary reaches one source node or another. The adversary is interested in no particular source node, any of them lead him to an event in the field. Therefore, if the adversary is able to reach the entry point of the mix network (i.e., the mix closest to the source) he will start to receive packets from the source node, thus revealing its location. The same applies to the exit point of the mix network and the protection of the base station. Finally, it is worth mentioning that the mix model provides attractive countermeasures against internal adversaries. They are successfully prevented from determining the source node and the base station unless they are precisely the entry or exit nodes of the mix network, respectively. The use of layered encryption prevents any compromised mix node or intermediate observer from obtaining information about the true data source since all these data are contained in the innermost layer. Also, the use of padding helps to hide the number of layers that were peeled by intermediated mixes. After traversing several mixes the messages are closer to the base station, thus padding prevents adversaries from learning the distance to the base station.

2.2.3 Onion Routing and Tor

Onion routing [108] is a low-latency anonymous communication system based on a network core composed of onion routers (OR), whose functionality is similar to Chaum's mixes. Indeed, onion routing is like mix networks except that the security of onion routing does not come from introducing significant delays to messages but from obscuring the route they traverse. Onion routers are connection-oriented devices, which means that once an anonymous connection (i.e., circuit) has been established through the network, the route remains unchanged for a given period of time. Circuits provide near real-time communications by multiplexing several connections in a single data stream using fixed-size cells. Moreover, circuits are established by means of a public-key layered data structure, called the onion. Each layer of the onion contains the cryptographic material needed to derive the symmetric keys used later during the data transmission phase by each of the onion routers of the circuit. The onion also tells each element of the circuit which is the next member. Once the circuit has bee established and onion routers have their session keys, application data are optionally sanitised to remove any sensitive information and then they are passed to the onion proxy which adds one layer of symmetric-key encryption for each of the onion routers in the path. Then, the entry onion router peels the outermost layer of encryption and sends the resulting message onto the next router. The process is repeated until the exit node removes the last layer and sends the data to the intended recipient. A simplified illustration of the onion routing architecture and its transmission process is provided in Figure 2.4a. Tor [43], the second-generation onion routing, added several changes to the original design, its new circuit setup process being the most relevant one. Instead of using an onion, the circuits are established incrementally, i.e., node by node, based on authenticated Diffie-Hellman key exchange. Moreover, the number of onion routers in a circuit is reduced from 5, in the original design, to only 3 as it was shown to provide reduced latency and similar security.

Onion routing reduces the overhead compared with mixes, principally for two reasons: data encryption and decryption is not based on public-key cryptography, and the core nodes are not required to temporarily store messages. Nonetheless, the computational and memory requirements are still costly for sensor nodes. Specifically, source nodes are required to be aware of the network topology as well as the public keys of each onion router to enable them to establish the anonymous path. Moreover, if the path is set by means of an onion, the source must perform several layers of public-key encryption containing the key seed material for each of the onion routers in the path. In the case of incremental path establishment, it implies that the source must contact the onion nodes one by one to make authenticated handshakes. This implies even more energy consumption because it requires the exchange of many messages, which is known to be much more power consuming than computations. Once the circuit has been established, the source node must apply as many layers of symmetric key encryption to the data messages as onion routers in the path. Later, each of the path members must decrypt the messages and multiplex several messages within a single link-encrypted transmission. In an attempt to further complicate traffic analysis,



(a) Onion Routing Communications

Node	**Requirements**		
	CPU		RAM
	Path setup	Transmission	
Sources	$N \cdot PK$	$N \cdot SK$	$M \cdot P_K + N \cdot S_K$ $+topo$
Onion nodes	$1PK$	$1SK + LE$ $(+PAD + PERM)$	$P_K + R \cdot L_K$ $+S \cdot S_K$

(b) Onion Routing Overhead

Figure 2.4: Onion Routing

packet padding and reordering is introduced by onion routing but Tor dismisses the idea because they introduce a significant cost and are still unable to yield effective resistance against various attacks. The base station might receive the data in clear text or encrypted with a shared key.

In Table 2.4b we summarise the computational and memory demands of onion routing schemes. The table considers both the path setup process, which only occurs occasionally, and the data transmission period. We place some operations in parenthesis those which are only performed by the original onion routing design and not by Tor. Extra terminology is defined: S_K is session key, LE and L_K are link encryption and link key, R is the number of neighbours an onion node shares links with, and S is the number of sessions an onion router handles at each given moment. During path setup, the source needs to perform a N public key operation and during data transmission these become symmetric key operations. Note that these values are per each single transmitted message. Moreover, the source nodes need to know the public keys of all M nodes in the onion network and somehow, the topology in order to be able to apply encryptions in the right order. Additionally, they need to store N session keys for each circuit, one for each of its nodes. We can assume that there is only one circuit per data transmission to the base station. The onion nodes only perform one decryption operation for each transmitted message. These decryptions are either based on public-key or symmetric-key cryptography depending on whether it is during the path setup or the data transmission. Moreover, onion nodes keep long-standing link encrypted connections with every other onion node. This is represented in the table as LE during data transmission. These together with padding and reordering are present during the whole communication, thus they could have also been included in the path setup, but it was done in this way for the sake of clarity. In terms of memory, onion nodes must keep their own public-key pair and as many link keys and session keys as there are neighbouring onion nodes and active circuits. Note that we used R instead of M to represent the number of link-key encrypted connection an onion node has but as an onion router should be able to connect to any other onion router in the network, this value may be equal to M. As a result, source nodes could be released from having to know the network topology. Again, we have considered the simplest case, where the source node does not use end-to-end encryption and the sink does not send responses back to the sources.

These schemes can be regarded as an evolution of the mix-nets approach in the sense that they reduce some of the tight requirements imposed by the original mix design. Despite the overhead reduction, onion routing solutions still present the same limitations with respect to the capabilities of the adversarial model considered in WSNs. The main drawback is that a local adversary will eventually identify the edges of the onion network. This issue allows him to identify the source nodes and the base station if messages follow similar or fixed routes to reach and leave the onion network. Therefore, the best strategy for an adversary is to reach entry or exit nodes and wait for messages to arrive. Overall, it can be stated that the edges of the onion network are the most critical points. This is also true if the adversary is capable of compromising nodes.

2.3 Decentralised Schemes

Contrary to centralised solutions, where the communicating parties are not involved in the anonymity network, in the solutions considered in this section all members collaborate to conceal the identities of other participants. In this way, there is more cohesion in the network, which positively affects the level of protection of the members since it is not trivial to identify the communicating parties from mere intermediaries. However, the elimination of a semi-trusted network core introduces new challenges. Note that some of these solutions are only partially decentralised because they rely on a central server, which is in charge of providing all the information necessary to communicate with other participants.

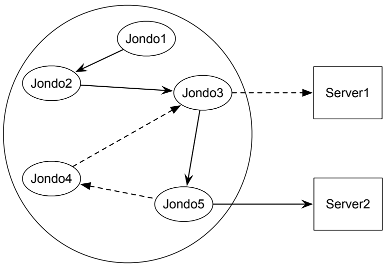
2.3.1 Crowds and Hordes

Crowds [109] is a partially decentralised solution where a set of geographically diverse users are grouped, and cooperate to issue requests on behalf of its members. Whenever a crowd member (i.e., a jondo) wants to send a message, it chooses a random jondo, possibly itself, to act as an intermediary. The receiving jondo decides, based on some biased probability, whether to forward the data to another jondo or to finally submit it to the destination. Subsequent requests from the same jondo and same destination follow the same path. Finally, the reply is sent back using the same path in the reverse order. Both requests and replies are encrypted using pairwise keys provided by a trusted authority, namely the blender, at the time jondos join the Crowd. This process is exemplified in Figure 2.5a, where $jondo1$ communicates with $Server2$ using three intermediaries while $jondo5$ issues requests to $Server1$ using only two relays. The first path, initiated by $jondo1$, is represented with ordinary arrows, while the other path is represented with dashed arrows. Interestingly, $jondo5$ is both the data source of one path and the last node in the other path. Hordes [70] is based on the Crowds model but its main contribution is the incorporation of multicast messages to reduce the latency and overhead on the return paths, i.e., from recipients to initiators. Additionally, it uses public-key cryptography to obtain the session keys from a trusted authority to be later used for message forwarding.

The Crowds model presents a low overhead when compared to other solutions. Instead of requiring computationally heavy mechanisms such as public-key operations, dummy traffic or padding, the Crowds is based on symmetric-key packet re-encryption, sender ID renaming, and random node selections[7]. Consequently, any intermediate jondo is only aware of the previous and next hop in the path and, from the receiver's perspective, the message is equally likely to have originated from any crowd member. Each member must perform one decryption and one encryption for every packet it forwards within the Crowd but, if it decides to submit the packet to the destination, it only needs to decrypt and forward it. In order to perform these operations, Crowd members must share keys with any other member. Therefore, the number of keys each node must store is dependent on the size of the network. Also, for every received message, the node changes the sender ID for its own and assigns an identifier to keep track of all messages belonging to that path. They must keep a translation table with as many records as the number of paths the node handles, because any subsequent packets from this connection will follow the same path.

Table 2.5b represents the number of operations and the amount of memory

[7] Additionally, the user might establish end-to-end encrypted channels to prevent en route eavesdropping by other Crowd members.



(a) Communications in the Crowd

Node	Requirements	
	CPU	RAM
Initial	$1SK$	$N \cdot S_K$
Intermediate	$2SK + 1RN$	$N \cdot S_K + R \cdot paths$
Final	$1SK + 1RN$	$N \cdot S_K$

(b) Crowds Overhead

Figure 2.5: Crowds

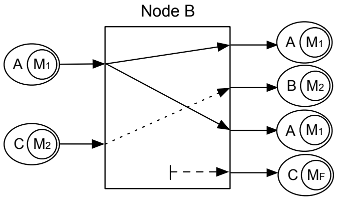
consumption introduced by the Crowds model to nodes with different roles in the network. Note that even when these roles are separated in the table, a node might have several roles at the same time. A similar table could have been constructed for Hordes but since we are not considering the communications on the return path in WSNs, this is not really useful. Additionally, in Hordes all participants hold the public key of the server, which is used to obtain a signed list of all other members and their public keys. Later each participant chooses a subset of jondos to use as message forwarders. The selected nodes receive a symmetric key encrypted with the node's private key. In this way, Hordes not only requires the storage of all participants' public keys but also the exchange and storage of session keys, which implies more computational operations and more memory consumption. For simplicity, we provide a single table corresponding to the Crowds solution. In this table, R represents the number of records in the translation table of an intermediate node. The remaining notation has already been introduced.

In general, the Crowds scheme imposes relatively low computational and memory requirements precisely due to the adversarial model under consideration. This solution provides a sufficient protection level against local adversaries which are able to observe the inputs and outputs of a single node but the attackers are considered to be static because of the geographic dispersion of the crowd members. This feature makes a big difference with respect to the WSN domain. The Crowds model considers a random but fixed path for all communications with a given server, however this involves a serious risk when the adversary can move towards the immediate sender of a packet. Likewise, by performing time-correlation attacks the adversary could determine the next hop in the path and after several hops he finally reaches the sink. Internal adversaries are partially countered by means of source renaming at every hop but the main drawback is that renaming also prevents the base station from learning the actual data source unless specified in the packet payload. Finally, this model provides no protection mechanisms against global adversaries, who can easily spot the data sources because crowd members start a transmission as soon as they have a request to issue. In other words, the transmission of real messages are not hidden by any means. Similarly, the base station can be easily detected by a global adversary because it is not part of the anonymity network.

2.3.2 GNUnet Anonymity Protocol

The GNUnet Anonymity Protocol (GAP) [14] was originally devised to provide anonymous file-sharing in peer-to-peer networks. GAP is based on the idea of making initiators look like mere intermediaries in order to hide their own actions. To achieve this, each node takes advantage of the traffic generated by other nodes but they also inject some baseline fake traffic in order to cover their own messages. Basically, GAP nodes perform the following actions: forwarding, renaming the identity of packets (i.e., indirection), injecting fake traffic, replaying messages several times, introducing short packet delays, and using message padding. Most of these actions are represented by different sorts of arrows in Figure 2.6a. An ordinary arrow means message forwarding, but these arrows may have forks which is represent the replay of packets to arbitrary nodes. Indirection is depicted by means of dotted arrows, while the short arrow starting from inside the node symbolises the injection fake packet injection.

The security of the GAP model is based on the idea that the more traffic a node transmits the more unlikely it is, to the eyes of an adversary, that a particular message was created by that node. In other words, a source node must route a sufficient number of packets from other participants so as to maintain an adequate protection level. Received messages can be either forwarded, indirected or dropped. Message forwarding implies no modifications to the message while indirection involves the modification of the sender address and thus the handling of subsequent packets belonging to that connection. However, in this analysis as we are considering traditional sensor networks where messages only flow from sensors to the base station, there is no need to handle replies, thus alleviating the problem of storing large translation tables. Only the forward path will be considered for the rest of this section.



(a) A GAP Node

Node	Requirements
CPU	$F \cdot FT + R \cdot PR$ $(+1RN)$ $+2SK$ $[+1SK]$
RAM	$N \cdot P_K + N \cdot S_K + T \cdot mess$

(b) GAP Overhead

Figure 2.6: GAP Scheme

Additionally, each node holds a public key that is used to establish encrypted links between nodes. Public keys are periodically propagated throughout the network. Also, both queries and data traversing the network are encoded using a particular scheme [15], which is similar to a symmetric-key encryption but it allows intermediaries to verify whether the encoded data matches a specific query or content. In this way, packets change their appearance at each hop but this also provides intermediaries with plausible deniability as they cannot decrypt what they are transmitting. This can also be considered a means of protection against internal adversaries. Finally, to further prevent the correlation between incoming and outgoing messages, short random delays are introduced and packets can be either forwarded or indirected to a random number of nodes.

The GAP model imposes extremely expensive requirements for hardware-constrained nodes, especially in terms of energy consumption since the network must maintain a baseline noise in the form of fake traffic and sensor nodes are battery-powered devices. The overhead introduced by this solution is summarised in Table 2.6b. Each node must contribute a given amount F of fake traffic to the network[8]. Moreover, for each received message a GAP node can decide to simply replay this message to a random number $R >= 0$ of nodes[9] or alternatively perform an indirection, which is represented within parenthesis. Also, after receiving a packet the node must decrypt it and then encrypt it with the key from the output link. Furthermore, source nodes perform an additional encryption operation, which is represented within brackets. From the point of view of memory, the node stores a number N of public keys, which are used to establish pairwise secrets for enabling link encryption with neighbouring nodes. Moreover, each node introduces a short random delay to messages which is translated into the

[8] This value F may vary on time depending on the network load.

[9] The range includes zero because the node may drop the message.

need of a buffer of a particular size T for allocating the messages. Recall that the variables represented on the table are node-dependent and may vary over time.

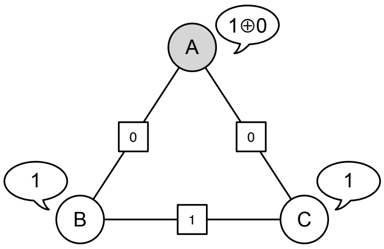
Both local and global observers can be countered by the GAP model since they cannot easily determine the source of messages due to the use of a baseline fake traffic that hides the occurrence of events. On the one hand, a local adversary does not gain any information by following all the messages since these might be fake traffic leading him nowhere. On the other hand, the global adversary is more difficult to deceive because in the presence of continuos events, there is an increase in the amount of traffic in that particular area compared to other more distant areas. Besides, internal adversaries are somehow, but not completely, countered due to the use of the encoding mechanism used for providing plausible deniability and the indirection mechanism. Finally, the base station can mimic the behaviour of ordinary nodes, replaying and sending traffic in order to remain hidden but it is likely that the area surrounding the base station still concentrates a larger amount of remote regions. In short, the presented mechanism might be useful for the protection of location privacy in WSNs but the overhead introduced will exhaust the battery of the nodes in a short period of time.

2.3.3 DC-nets and Herbivore

The Dining Cryptographers (DC) scheme [27] allows a group of users to share information while hiding the actual sender of messages even to other protocol participants. To this end, each member needs to share a secret bit with any other participant. For example, in Figure 2.7a, node B shares a 0-bit with node A and a 1-bit with node C. Also, the participants perform the sum modulo 2 (i.e., logic XOR) of their shared secrets. Subsequently, the obtained result is broadcasted to the rest of participants unless the participant is willing to communicate data to the rest of members, in which case it shares the inverse of the result (see node A in Figure 2.7). The final result is obtained by performing the XOR of all contributions. Each protocol execution is called a round.

The idea behind this scheme is that the final result must be zero if nobody (actually, any even number of simultaneous senders) has transmitted because each secret is used twice, and one if someone inverts de result[10]. Provided that the initial shared bits are secret, there is no way to determine the actual sender. Although the original protocol considers the transmission of a single data bit, the

[10] The inverse of any bit value is that same bit value xor-ed with 1.



(a) DC-nets Overview

Node	**Requirements**
CPU	$2 \cdot XORs\ (+\ INV)$
RAM	$[2\ to\ N-1]$ bits
Other	Topology restrictions, tight synch, error prone, simultaneity

(b) DC-nets Overhead

Figure 2.7: DC-Nets Scheme

DC scheme can be easily extended to transmit string messages by sharing random numbers instead of random bits. This modification enables the transmission of encrypted messages so that the actual recipient is the only entity capable of determining whether that protocol round conveyed a real message. Thus it provides unobservability of both senders and recipients. The following analysis focuses on the bit-based version but it could be directly extrapolated to the extended version.

The application of the DC-nets model in WSNs has several impediments. One of these limitations is that sensor networks communicate wirelessly, which is a highly unreliable medium. The DC protocol is extremely vulnerable to noise and a single erroneous bit leads to undesirable results. Additionally, provided that participants' contributions must be broadcast[11] simultaneously in order to allow the XOR of their signals, the sensor nodes running the protocol are required to be tightly synchronised and within the transmission range of the other members. This suggests that the data recipient must be either one of the DC participants or an external observer within the communication range. Consequently, only neighbouring sensor nodes can run the protocol or they must carefully adjust their transmission power, however, this would deplete their limited batteries in a short period of time. As proposed by Herbivore [48], the participants could be hierarchically arranged in order to reduce the complexity of the system. This arrangement would allow sensor nodes to reduce their transmission power but it also introduces more synchronisation problems and increased delivery delays to data packets.

Additionally, there are high memory requirements in the DC model associated with the key sharing process because of the continuous protocol rounds. Two

[11] There are other potential communication techniques besides broadcasting but they imply an increase in the number exchanged messages.

potential solutions exist for the provision of keys: either sensor nodes are preloaded with sufficiently large one-time keys, or they share short keys which are periodically updated by means of a pseudorandom function. In the former case, given the memory limitation in sensor nodes, the shared keys will rapidly expire. In the latter case, the memory cost is traded by computational operations. In any case, the overhead introduced is directly dependent on the topology of the network. A ring topology, such as the one presented in Figure 2.7a, requires each node to share 2 random bits, with the right and left participants. On the other hand, in a fully-connected graph each participant shares one bit with every other participant, which adds up $N-1$ random bits, where N is the total number nodes. Note that these values are for a single protocol round (i.e., for the transmission of a single bit). Moreover, a protocol round occurs even if no participant is willing to transmit otherwise an adversary would identify which nodes are interested in transmitting. Clearly, this implies a high waste of bandwidth and energy because of the continuous flow of messages.

Another substantial problem has to do with simultaneous communications. The DC model does not allow various data senders at a time because their messages would collide. This issue highly constrains the usability and nature of sensor networks, which were conceived to provide a highly distributed sensory system. This problem might be reduced by using a slot reservation protocol as proposed by Herbivore, however, this introduces more messages and thereby more energy waste. Moreover, this countermeasure cannot solve the increased delivery time in the communications, especially when the sensor networks under consideration are extremely large with a substantial amount of potential data senders. A summary of these and other features constraining the application of this model to WSNs are presented in Table 2.7b, where INV refers to the inversion of the contribution. For every protocol round, each node is required to perform only two simple XOR operations and, optionally, an additional one if they want to transmit data. In terms of memory requirements, depending on the connectivity of the network, a single protocol round requires from 2 to $N-1$ secret bits.

Although the computational overhead introduced by the DC-net scheme is rather inexpensive even for sensor nodes, the memory requirements, topological restrictions and the disruption of simultaneous event notifications preclude their application to WSNs. Nonetheless, the model is effective in the protection of location privacy because it hides the original data source to all participants and also external (local or global) observers. This could result in a problem for the base station which is unable to identify the data source unless the extended protocol is used. To this end, the source node would send both the event data and its identifier in an encrypted form so that only the base station knows the original sender. Therefore, the location of the data source is protected from disclosure to any other participant including internal passive adversaries, which are unable to determine the original data sender unless they collude. As a matter of fact, a collusion is successful only if all nodes sharing keys with the potential source node collude which is highly unlikely.

2.4 Evaluation

Previous sections have delved into several features from centralised and decentralised anonymous communication systems that need to be further analysed. This section is intended to provide this final discussion while outlining the most important aspects of this chapter.

As for the case of centralised solutions, these can be regarded as black box devices where the data sources stand on one side and the data recipients on the other. The communications originating from various sources change their appearance, are delayed or mixed within the network, but still the presence of incoming and outgoing messages is evident. In these settings, both source nodes and the base station are clearly exposed to a global observer, simply because they are not part of the network core and thus their actions can be easily detected, which implies the disclosure of their location. Contrarily, local and internal adversaries are placed somewhere within the network core and, in consequence, they cannot identify the communicating nodes so easily. These adversaries rely on a partial view of the communications but depending on their location they might be more likely to uncover the senders and recipients. The entry and exit points of centralised systems are especially sensitive since at these areas the adversary is capable of distinguishing the source nodes and the base station unless packets use different routes to reach that point.

In particular, single-proxy schemes are very lightweight because they are primarily based on source renaming at a single intermediate point. However, this together with the potential use of payload encryption for eavesdropping prevention, can protect neither from the trace-back attack performed by local adversaries nor from compromised proxy nodes because they can retrieve the data source from the packet.

Mix-based designs depict a rather different situation. The overhead imposed by mix nodes is significant, not only because it demands the use of public-key cryptography but also because the source node must perform as many of these operations as nodes in the communication path each packet traverses. Additionally, the layered encryption implies the knowledge of the public keys of every mix node and the topology of the network to perform the encryptions in the right order. Moreover, mixes introduce large message delays, which are not suitable for time-critical applications, like critical infrastructures monitoring. Regarding the privacy protection, mix cascades present the same problem concerning local adversaries, which are able to follow the paths of messages since they are fixed and they follow any received packets regardless of the appearance or timing. Yet, the free-route selection proposed by mix-nets provides some protection means against local adversaries but it might still be insufficient since they can eventually reach either edge of the mix network. From these positions, local attackers are much more likely to succeed. Similarly, internal adversaries who are at the edge of the network are capable of uncovering the communication endpoints. However, the use of layered encryption prevents intermediate nodes in the path from uncovering the data source. More precisely, intermediate nodes are only aware of the previous and next hop in the path.

Finally, onion routing solutions reduce some of the computational restrictions imposed by mixes by introducing the path setup process, which allows the establishment of session keys that are later used during the data transmission process. Also, these schemes reduce the delay introduced at every hop by multiplexing the communications of various data sources on a single stream. Although the overhead is reduced, it still demands layered cryptography and great memory requirements. Anyway, onion routing schemes present the same problems when countering the typical adversaries considered in sensor networks.

As for the case of decentralised approaches, their aim is to prevent the aforementioned problems at the edges by making all participants part of the system. In other words, any member of the system is potentially a data source as well as a data forwarder. This implies that it is not trivial for global observers to determine the communication endpoints and it also introduces the opportunity to more sophisticated internal attacks.

The Crowds schemes do not sufficiently protect against global adversaries because the data recipients are not part of the network and the data senders, start new paths for new data connections, thus altering their behaviour and becoming an easy target. Moreover, in order to keep a low overhead, these solutions do not introduce protection mechanisms such as dummy packet injection, which might be helpful against both global and local adversaries. Local adversaries can also trace back sources and the base station because the paths are static once created to reduce the chances of internal adversaries. Internal adversaries are countered only slightly because, even if some protection means are placed (i.e., renaming), data sources can be easily detected by its neighbours for the same reason global adversaries can identify them.

To the contrary, GAP and DC-nets offer attractive safeguards, which improve the level of protection against the various types of adversaries considered in the WSN domain. However, these safeguards imply a significant increase in the number of messages being transmitted, replayed or forwarded, which results in an unaffordable energy waste for battery-powered devices. Additionally, the DC-nets model presents extra limitations in terms of memory requirements, network topology restrictions, and also the inability to handle simultaneous data sources, which further precludes its application to the location privacy problem in WSNs.

A visual summary of this discussion is presented in Table 2.2, where the up and down arrows roughly indicate the overhead introduced and the impediments presented by these systems with respect to their applicability to WSNs. The tick, cross and approx symbols ($\surd$, $\times$ and $\approx$) represent whether these solutions can provide, are not able to provide or could provide some protection against the three adversarial models considered in WSNs.

In general, we can state that centralised approaches are less suitable for the protection of location privacy in WSNs than decentralised approaches given the highly distributed nature of these networks and their particular communication pattern. The typical many-to-one communication model makes it difficult to hide the location of the base station and the source nodes when they are located outside the limits of the centralised network core. A local adversary can eventually determine the entry points of the network core while a global adversary can directly identify the source and destination of messages. Therefore, decentralised approaches are more appropriate as they integrate all the nodes within

	Overhead	**Adversary**		
		Global	Local	Internal
Single-proxy [10]	↓↓	×	×	×
Mix-nets [26]	↑↑↑	×	×	✓
Onion routing [108]	↑↑	×	×	✓
Tor [43]	↑↑	×	×	✓
Crowds [109]	↓	×	×	≈
Hordes [70]	↓	×	×	≈
GAP [14]	↑↑↑	✓	✓	✓
DC-nets [27]	↑↑↑	✓	✓	✓
Herbivore [48]	↑↑↑	✓	✓	✓

Table 2.2: Suitability of Traditional Systems

the anonymising solution, thereby hindering the identification of the current participants to adversaries with either local or global eavesdropping capabilities. However, not all decentralised solutions are capable of providing suitable protection.

Although some solutions are sufficiently lightweight to run in a sensor node, we have shown that the real weak point is that they do not fit the requirements and the adversarial models considered in the sensors domain. Similarly, another group of solutions are suitable for the protection of location privacy in WSNs but they are rather expensive in terms of computational, memory, and battery requirements or they present additional limitations. However, the analysis of these solutions has provided us insight into a variety of mechanisms and techniques which can be applied to the sensors' domain to preserve location privacy.

Chapter 3

Analysis of Location Privacy Solutions in WSNs

Anonymity systems for traditional communication networks have been studied and it has been concluded that these solutions are not practical for the features, restrictions and attacker models considered in WSNs. As a result, this chapter delves into the various solutions devised to protect both source- and receiver-location privacy. The idea is to gain insight into the techniques used by different solutions and learn from studying their advantages and disadvantages in order to come up with new and improved solutions capable of solving some open issues in the area.

The contents of this chapter are organised following three criteria, which are (a) the asset or information the solution aiming to protect, (b) the capabilities of the adversary to be countered, and (c) the techniques used by the solution. In Section 3.1, we describe and analyse solutions dealing the protection of the identities of the nodes, which are carried within the packet headers to enable the routing of data through the network. This is the first step to location privacy protection. Next, Section 3.2 and 3.3 concentrate on the obfuscation of traffic patterns in order to protect the location of the data sources and the base station in the presence of different adversarial models. Finally, we present a complete taxonomy of location privacy solutions and some research gaps that we tackle in the following chapters.

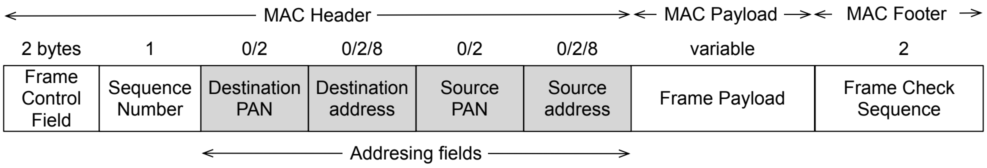
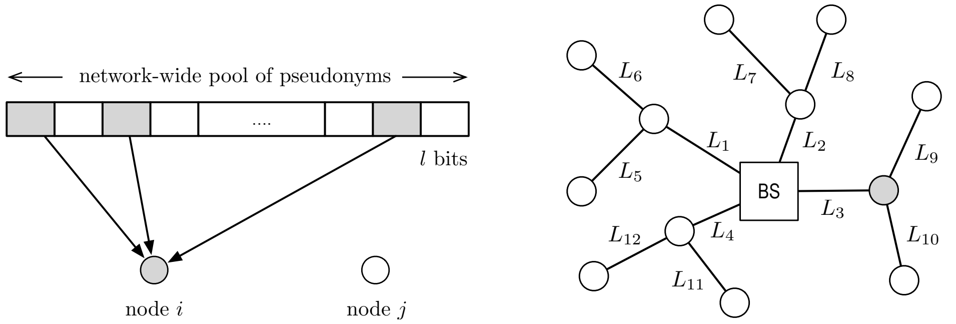


Figure 3.1: General IEEE 802.15.4 MAC Frame Format [135]

3.1 Node Identity Protection

Despite the use of cryptographic mechanisms to protect the payload of data packets, there is much relevant information contained in the packet headers that is available to anyone eavesdropping on the wireless channel. Packet headers consist of various data fields containing, among other things, the identifiers of the data sender and the destination (see Figure 3.3 for the structure of a typical WSN frame). These data are sent in clear text because any intermediate sensor node must be capable of using packet header information to perform routing tasks. Therefore, an attacker can, after a sufficient amount of time capturing network traffic, elaborate a map of the network relating node identifiers to locations in the field. Being in possession of such a network map, a local attacker may simply wait next to the base station for incoming messages because all packets are addressed to this single location. Upon the reception of a packet, the adversary can retrieve the identifier of the data source and, by using the map, he can translate the identifier into a physical location, where the event occurred.

Several techniques have been proposed in the literature to provide node anonymity. Most of these solutions are based on the creation, distribution, update, and use of pseudonyms, which are intended to hide the true identifiers of the nodes. Persistent pseudonyms provide no means of protection in the long term as they become the new identifiers of the nodes and thereby the attacker is able to easily correlate a node to a pseudonym. Therefore, pseudonyms are only effective if they are periodically updated, that is, pseudonyms must be dynamic if they are to provide node anonymity. Some authors have approached the management of pseudonyms by means of pools of pseudonyms while others have turn to cryptographic mechanisms for the same purpose. Note that most of the solutions fall into the second category since the use of cryptographic techniques for the creation of pseudonyms have several benefits over the use of network pools. Next, we review these solutions in detail.



(a) Distribution of pseudonyms in SAS (b) Label assignment in DCARPS

Figure 3.2: Pool-Based Approaches

3.1.1 Pool of pseudonyms

Misra and Xue [92] were the first authors to provide a set of solutions for node identity protection based on the use dynamic pseudonyms. The first of their solutions, called the Simple Anonymity Scheme (SAS), is based on a network-wide pool of pseudonyms which are distributed among the sensor nodes. The base station divides the pool into subranges of l bits and provides each node with a random set of these subranges (see Figure 3.2a). Moreover, the base station stores correspondence between the identity of each sensor node and its subranges in order to figure out the correct decryption key for received messages. After deployment, each node builds a pseudonyms table where it stores the pseudonym ranges and secret keys used for communicating with its near neighbours. In each row of the table, the node keeps two ranges of pseudonyms for traffic coming from and directed to a particular neighbour. When the node wants to send data to a neighbour, it selects a random value from the range of pseudonyms belonging to that node and concatenates the index of the row from where it picked the pseudonym. The recipient node checks whether the received pseudonym belongs to the incoming range corresponding to the given index and, that being the case, it uses the shared key to decrypt the message. The principal limitation to SAS is the large memory space necessary to store a sufficiently large pseudonym space. Note that each sensor node is assigned several ranges of l bits from a pre-established pool of pseudonyms, and uses two ranges for each of its neighbours. This imposes a high memory overhead for hardware-constrained devices, especially in densely populated networks.

Nezhad et al. [93, 94] proposed a label switching protocol for providing node anonymity as part of their Destination Controlled Anonymous Routing Protocol for Sensornets (DCARPS). After each topology discovery phase, the base station is aware of the location of the sensor nodes and is able to build an updated map of the network. This information is used to assign labels (i.e., identifiers) to each and every network link, as depicted in Figure 3.2b. These labels serve as pseudonyms and whenever a node has to send a packet to the base station, it uses the label assigned to the link connecting it to a neighbour that is closer to the base station[1]. Upon the reception of the packet, the neighbour node, checks whether the label corresponds to one of its input labels. If the label is known to the node, it replaces the input label with its own output label. For example, the grey node in Figure 3.2b checks whether an incoming message has either label L_9 or L_{10} and, in the case it does, it forwards the packet after changing the original label with L_3. The main drawback to this labelling solution is that it is not sufficiently dynamic. Labels are only modified sporadically, after a topology change has been discovered in the network, which gives the attacker the opportunity to observe the same labels for large periods of time. This allows the attacker to correlate labels with specific nodes, thus completely compromising anonymity.

3.1.2 Cryptographic pseudonyms

The second solution by Misra and Xue [92] was the first one to use a cryptographic scheme to preserve the identity of the nodes. This solution is intended to reduce the amount of memory needed to handle the ranges of pseudonyms in SAS at the expense of increased computational overhead. The Cryptographic Anonymity Scheme (CAS) uses a keyed hash function to generate the pseudonyms. Before the deployment of the network, each node x is assigned a pseudo-random function f^x and a secret key K_{sx} as well as a random seed a_{sx} for communicating with the base station. After that, each pair of neighbours agree upon a random seed and a hash key generated using the pseudo-random function f^x. This information together with a sequence number seq are stored in a table which is used to generate the pseudonyms during the data transmission period. Whenever a node x wants to communicate with the base station, using node y as intermediary, it creates a message $M = \{sID, rID, EncryptedPayload, seq\}$, where sID consists

[1] Each node is considered to use only one path to send data to the base station for simplicity reasons. There is probabilistic version of DCARPS where nodes select, for each packet, a random node from all possible communication paths.

of the concatenation of the index of node y in the table and the hash function of the bitwise XOR operation of the random seed shared with the base station and the sequence number keyed with the secret shared with the base station (i.e., $sID = I_y||H_{K_{sx}}(a_{sx} \oplus seq_{xy})$). The contents of rID are very similar except that instead of using the key and seed shared with the base station it uses the ones shared with the neighbour y. Clearly, the first field is used for identification with the base station and the second is used for identification with the next hop in the communication. This scheme is more memory efficient but it introduces a relevant computational cost, not only to intended recipients but also to the remaining neighbours which need to compute a keyed hash value before discovering that the packet is not addressed to them.

The CAS scheme ensures that an external observer cannot learn the real sender (or recipient) of a message by simply observing the identifiers contained in the packet headers. The authors assume that an attacker cannot compromise the secrets shared between the nodes. For example, if an attacker captures a node, he learns all past, present and future pseudonyms. To reduce the impact of secrets being compromised, Ouyang et al. [99] propose two methods based on keyed hash chains. The Hashing-based ID Randomisation (HIR) scheme, uses the result of applying a keyed hash function to the true identifier of the node as pseudonym. More precisely, after the topology discovery process, sensor nodes determine which neighbours are closer to (uplink) and which are farther from (downlink) the base station, and share pairwise keys with them. Then, sensor nodes create a table that includes, for each link, the keyed hash identifier of the uplink node of that neighbour[2]. After the transmission or reception of a message on a particular link, the node rehashes the value contained in the table to generate a fresh pseudonym. Additionally, packets convey another identifier used for the base station to be able to identify the original data source. This value is also an element of a hash chain keyed with a secret shared with the base station. Since hash values are assumed to be non-invertible, this solution provides backwards secrecy, that is, an attacker compromising the node or the secrets cannot retrieve previous identifiers.

However, if the adversary compromises the key used for the hash functions he can easily generate future pseudonyms since he only needs to rehash the last

[2] If the node itself is the uplink of its neighbours, it stores the hash value of its own identifier keyed with the secret shared with the corresponding neighbour.

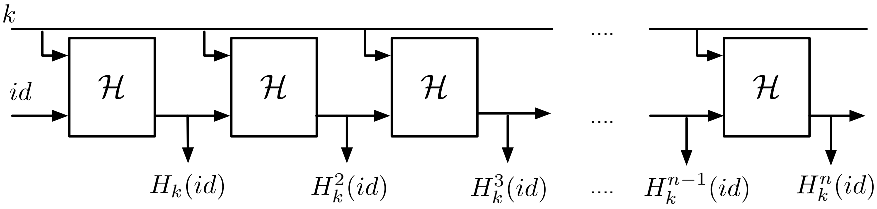


Figure 3.3: Keyed Hash Chain Generation

values used by the node. The second solution by Ouyang et al. [99] attempts to further reduce the risk of secrets being compromised. Instead of creating the identifiers on the fly as they are needed, in Reverse HIR (RHIR), the nodes first create the hash chain, store it locally, and then use the elements of the chain in reverse order. Once a pseudonym has been used, it is no longer needed and it can be deleted from the memory. Also, even if the attacker obtains the secret key used to create the hash chain, he cannot generate any fresh pseudonyms since he cannot invert a hash. The main drawback to this solution with respect to the previous one lies in the need for increased memory space to accommodate a lengthy hash chain.

Later, Jiang et al. [59, 131] introduced the Anonymous Path Routing (APR) protocol. One of the elements of this scheme, namely the anonymous one-hop communication, introduces an enhancement that improves the resilience against secret compromise attacks compared to previous solutions. Similar to the solutions by Ouyang et al., in this scheme each node creates a table to keep the uplink and downlink identities of each neighbour. These hidden identities are calculated by hashing the values of the secret keys, identities, a sequence number and a nonce shared by the nodes. The novelty of this approach is that both the shared keys and the hidden identities are updated (i.e., rehashed) after each successful transmission between neighbouring nodes.

The same idea has been developed by Chen et al. [30, 31] in the Efficient Anonymous Communication (EAC) protocol. Before the deployment of the network each sensor node is preloaded with two hash functions, a secret key and a random nonce shared with the base station. These data are used to generate a pseudonym that is included in packets addressed to the base station in order to allow the identification of the data source. The pseudonym is updated for every new packet by applying one of the preloaded hash functions to the current identifier xor-ed with the random nonce. The problem with this scheme is that, after deployment, each node exchanges their preloaded information with its neighbours in order to generate and update pseudonyms for one-hop communications. This information includes the keys and nonces shared with the base station, which allows any node to determine whether the true source of the packet is a neighbouring node. Indeed, a node could even impersonate any of its neighbours.

None of these schemes can successfully protect the system from attackers who are able to capture a node and access its internal memory. When a node is compromised, its secrets are exposed and the adversary retrieves all current pseudonyms and is able to generate all future pseudonyms. Notwithstanding, we acknowledge that coming up with a solution capable of dealing with this type of threat is rather challenging. Some kind of node revocation mechanism would be necessary to diminish this sort of problem.

Finally, it is important to highlight that node anonymity is only a first line of defence to preserve location privacy. This problem is a huge challenge due to the resource limitations of the scenario and the peculiar communication model of these networks, which together allow a skilled adversary to perform more sophisticated traffic analysis attacks to determine the location of the nodes of interest to him. In the following sections we present and analyse the most important solutions that have been developed to diminish the threat of different types of adversaries. The exposition will be based on the capabilities of the adversaries, more precisely on their eavesdropping power and their ability to capture nodes.

3.2 Source Protection

Source-location privacy refers to the ability to protect the location of the sensor nodes reporting event data to the base station. More precisely, source-location privacy is intended to prevent an attacker from finding the physical location of the events being monitored by the network since they may be related to individuals or valuable resources.

This problem has drawn the attention of the research community due to the challenging nature of the scenario. Many solutions have been devised for countering passive adversaries with a local or a global view of the communications but only a few authors have concentrated on the threat of internal attackers.

3.2.1 Local Adversaries

A local adversary can only monitor a small portion of the network, typically the equivalent of the hearing range of an ordinary sensor node, that is why they are usually referred to as mote-class attackers. Therefore, they must turn to moving in the field following packets until they find the data source. This strategy is called a *traceback attack* since the adversary attempts to reach the target by moving along the path of messages from the source to the base station in reverse order.

A traceback attack is successful in typical WSNs because the packets transmitted by a particular node tend to follow the same path over and over again. Consequently, most of the solutions to this problem are based on the randomisation of routes (i.e., using different paths for different packets) to hinder traceback attacks. The goal is to mislead the adversary in order to increase the *safety period*, that is, the number of packets sent by the source node before the attacker reaches it. The application of route randomisation protocols come at the cost of increased latency, higher packet loss probability, and most importantly, increased energy waste. The research community has struggled to find the right balance between network performance and privacy protection.

Below we analyse a number of solutions falling into some of the following categories, namely undirected random paths, directed random paths, network loops, and bogus traffic. In the first category, we include solutions where the communication paths are not clearly guided by a mechanism to improve the safety period while the solutions in the second category introduce a technique to direct the random walks. The solutions in the third category use a strategy based on the creation of loops of fake messages in order to deceive the adversary into believing he is following a real path. Bogus traffic has also been used in different ways to protect the data sources. Note that some solutions may belong to more than one category.

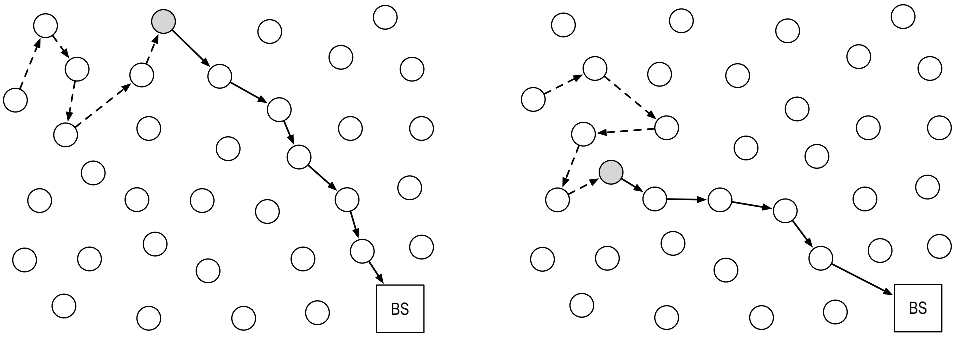
Undirected Random Paths

The first solution to provide source-location privacy was devised by Ozturk et al. and is called Phantom Routing [100]. This scheme results from the analysis of two widely used families of routing protocols in WSNs, flooding-based and single-path routing protocols. Surprisingly, both provide the same privacy protection level although it may seem that an attacker could be confused in a flooding by the number of messages coming from all directions. However, the attacker only needs to pay attention to the first message it observes since this is following the shortest path. Besides that, a baseline flooding wastes a significantly greater amount of energy compared to single-path routing but the latter is less robust to packet loss. Probabilistic flooding tries to find a balance between reliability and energy efficiency by making sensor nodes flood messages with a given probability. As a side effect, this approach reduces the likelihood of the attacker reaching the data source.

Based on the previous analysis, Phantom Routing proposes making each packet undergo two phases, a walking phase and a flooding phase. In the walking phase, the packet is sent on a random walk for h hops until it reaches a node, which is called the phantom source. Then, in the next phase, the phantom source initiates a baseline or probabilistic flooding, which eventually delivers the packet to the base station. This two-phase process is repeated for each new message thereby selecting random phantom sources. Having different phantom sources implies that messages traverse different paths, which reduces the location privacy risk for the actual data source. Later, a new version of protocol, called Phantom Single-Path Routing, was proposed in [60]. This variant replaces the flooding in the second phase by a single-path routing, which results in even longer safety periods due to the fact that the adversary misses some of the single-paths coming from different phantom sources. Figure 3.4 depicts the transmission of two messages using the Phantom Single-Path Routing protocol, where dashed arrows represent the walking phase and the ordinary arrows represent the single-path phase. The grey node in both subfigures represents the phantom source for each transmission.

The main limitation to Phantom Routing protocols is in the walking phase. Pure random walks tend to stay close to the source node and the definition of a large value for h does not solve the problem. Indeed, a larger value of h does not provide a direct improvement in the safety period, it only increases the energy waste. This problem is represented in Figure 3.4, where phantom sources are within a distance of two or three hops regardless of the definition of a 5-step random walk.

To reduce the concerns about pure random walks staying close to the source node, Xi et al. [148] propose GROW, a two-way greedy random walk. The idea



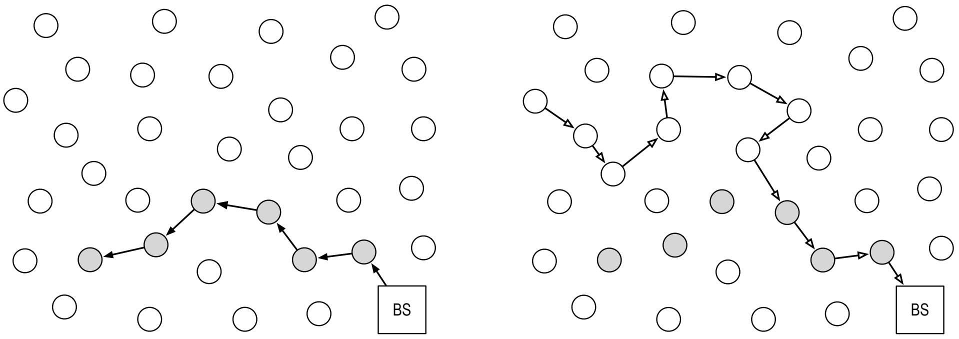
(a) First transmission (b) Next transmission

Figure 3.4: Phantom Single-Path Routing with $h = 5$

behind GROW is that using random walks is desirable for protecting source-location privacy because routing decisions are made locally and independently from the source location. However, using pure random walks as the only routing mechanism is impractical because the average delivery time of messages goes to infinity. GROW exploits the fact that the probability that two random walks will not intersect decreases exponentially in time [125]. First, it creates a permanent path of receptors by transmitting a special packet on a random walk from the base station. Then, the source node sends all subsequent data packets on a greedy random walk that will eventually hit a node from the path of receptors. From there, the packet is forwarded to the base station following the established path in reverse order. This process is illustrated in Figure 3.5. The protocol is said to be greedy because it uses a Bloom filter[3] to store previously visited nodes in order to extend as far and as quickly as possible. Despite being designed as a greedy algorithm, one of the main limitations of GROW is the substantial delivery time of the packets.

Cross-layer routing [128] was designed to further mitigate the problem of random walks staying close to the data source. This approach is basically a Phantom Routing that hides the walking phase by routing data using the data link layer. Beacon frames are periodically broadcast to inform about the node presence and other network related parameters. Additionally, frame payload can be cryptographically obscured which allows sensor nodes to convey event data in securely. Since beacons are transmitted regardless of the occurrence of events, the

[3] A Bloom filter [18] is a simple data structure used for representing in a memory-efficient way (as a bit string) a set of elements and for supporting queries about whether an element belongs to the set, or not.

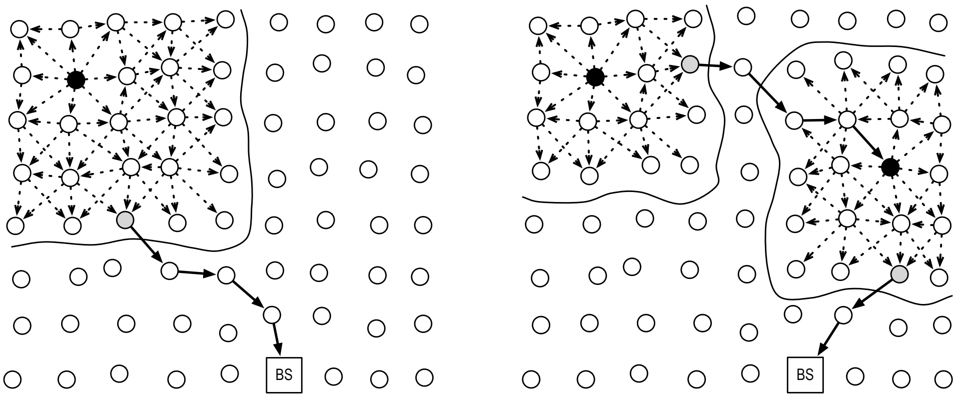


(a) Path of receptors (b) Data transmission

Figure 3.5: Operation of the GROW Scheme

attacker is unable to distinguish legitimate beacons from those containing event data. At the end of the walking phase, event data reaches a pivot node where the information is extracted and sent to the base station using the implemented routing protocol. The pivot node is chosen by the data source at random from all its neighbours at h hops of distance. The operation of the protocol is depicted in Figure 3.6a, where the dotted arrows represent the routing phase at the data link layer, solid arrows represent the transmission of messages at the routing layer, and the black and grey circles represent the data source and the pivot node, respectively. This solution provides perfect privacy for all attackers within the beaconing area as long as they are not close to the pivot node. Also, since the routing layer mechanism considered by the authors is a single-path protocol the attacker only gains some information if he is on the path from the pivot node to the base station. The main limitation to this approach lies in the tradeoff between the level of protection it can provide and the delay introduced by large beaconing areas. Beacon frames are periodically sent out at intervals ranging from milliseconds to several hundreds of seconds. Therefore, the larger the beaconing area is the better the protection but also the longer the delay.

As the data travels from pivot nodes to the sink using a single-path strategy, choosing nearby pivot nodes very often allows an attacker to determine and reach the edge of the beaconing area. Also, due to the important tradeoff involving the size of the beaconing area, the network administrator may turn to small values for h in order to boost the delivery time. This implies that pivot nodes will be close to the original data source (i.e., same problem as with the original Phantom Routing) and even if there is no evidence of messages leading to the target, the



(a) Single cross-layer approach (b) Dual cross-layer approach

Figure 3.6: Cross-Layer Routing Schemes

uncertainty region is considerably reduced. Therefore, an attacker can turn to a systematic field inspection to find the source node with no great effort. A double cross-layer solution is proposed by Shao et al. to further enhance location privacy in these circumstances. In this version of the protocol, instead of sending the data directly to the base station, the pivot node sends the data to another randomly chosen node using the routing layer. Then, this random node chooses a new pivot node and starts a second beaconing phase. Thus, the attacker cannot easily reach the edge of the beacon area to which the original data source belongs. The dual cross-layer approach is represented in Figure 3.6b.

Based on the same idea of hiding the walking phase, Mahmoud and Shen propose creating a cloud of fake traffic around the data source to hinder traceback attacks [83, 84]. During the network setup, the base station floods the network with a discovery message in order to allow sensor nodes to learn the shortest path to the base station as well as the nodes in that route. Then, sensor nodes choose a group of nodes at different distances to become fake source nodes, similar to phantom sources or pivot nodes. Finally, each node groups its immediate neighbours in such a way that the members of each group are not contiguous so as to allow each group to send packet in different directions. During the data transmission phase, for each message the source node chooses one node Fs from its list of fake sources and sends the message to the group where there is a member which knows how to reach Fs. As the packet travels to the fake source, it generates fake traffic to cover the route. A node from the addressed group generates fake traffic if it is not in the direction of the fake source. In that case, the node chooses one of its groups at random and sends a fake message that lasts for h hops. Consequently, if groups are carefully chosen, traffic flows in any possible direction, generating clouds with dynamic shapes.

Compared to the cross-layer scheme, the main limitation to the cloud-based approach is that the clouds of fake messages consume substantially more energy than beacon frames, which are present even if there is no event data to transmit. On the other hand, routing data in the link layer is very slow and introduces significant delays but it is an interesting countermeasure when there are high privacy demands.

Directed Random Paths

Instead of simply sending packets at random, some authors have proposed using mechanisms to guide the walking phase. By having a walking phase governed by certain parameters, either the packet delivery time is reduced or the privacy protection level is increased, or both.

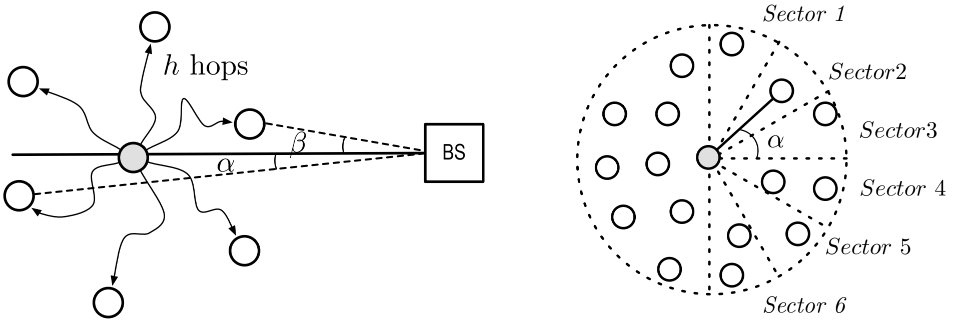
The first solution to have included a mechanism to guide the walking phase is Phantom Routing itself [100]. The authors suggest changing the pure random walk in favour of a directed random walk. To that end, each node separates its neighbours into two groups depending on whether they are in the same direction or in the opposite direction to the base station. Thus, during the walking phase, the next hop in the path is still selected uniformly at random but only from the set of nodes in the direction of the base station. By introducing this simple mechanism they prevent packets from looping in the vicinity of the source node and thereby achieve a similar safety period while reducing the energy waste.

Yao and Wen devised the Directed Random Walk (DROW) in [152]. The idea behind this solution is quite simple, any sensor node having a data packet to transmit must send it to any of its parent nodes (i.e., a node closer to the sink) with equal probability. This applies to both data sources and intermediaries. The level of protection provided by DROW is therefore highly dependent on the connectivity of the network. A path with a limited number of neighbours implies a short safety period since most of the packets follow very similar routes to the base station. In 2010, Yao alone published another paper describing the Directed Greedy Random Walk (DGRW) [151]. This solution is a mere copy of DROW with a different name. Also, the Forward Random Walk (FRW) scheme by Chen and Lou [29] does exactly the same thing. However, the authors argue that this solution cannot obtain a high level of protection and it would be necessary to inject dummy messages in the network to reduce the chances of the adversary.

Later, Wei-Ping et al. [144] observed that one of the most critical factors during the walking phase period is not the length of the walk but its inclination. Long random walks do not necessarily increase the safety period unless the phantom sources are placed in a safe location to initiate the routing phase. A location is considered to be safe if it is not close to the straight line between the data source and the sink. The reason is that if phantom sources are close to this line too often, the single paths originated by them will be very similar to each other and thus the attacker has more opportunity to overhear packets. This problem is depicted in Figure 3.7a, where the curly lines represent directed random walks from the source node to the phantom sources and the dashed lines represent the single-path routing phase. To prevent this situation, Phantom Routing with Locational Angle (PRLA) prioritises the selection of phantom sources leading to larger inclination angles. More precisely, a sensor node assigns its neighbours forwarding probabilities based on their inclination angles in such a way that neighbours with larger angles will be more likely to receive messages. After h hops, the node receiving the message becomes a phantom source and finally sends the packet to the base station using the shortest path. By using this strategy, the authors manage to reduce the number of hops necessary in the walking phase while keeping an adequate safety period. A major downside to this work is that it is not fully clear how the nodes obtain the inclination angles[4] of their neighbours without built-in geolocation devices or directional antennas.

Wang et al. [140] propose Random Parallel routing, which assigns each sensor node n parallel routing paths to the base station. Messages are evenly distributed to different paths in such a way that the adversary traceback time is the same at any path. Also, the paths must be sufficiently geographically separated in order to prevent the attacker from overhearing packets from various paths. The underlying idea is that if the adversary chooses one of the paths he is forced to stay on that single path. This improves the safety period, which is now equivalent to the sum of all the parallel paths. More formally, let L_i be the length of each of the paths and let p_i be the probability of choosing the path i as the transmission path. Then, the traceback time for an attacker (i.e., the safety period) is equal to

[4] The authors claim that the inclination angle of neighbours is calculated in terms of the number of hops. Nonetheless, two nodes at the same distance have different inclination angles.



(a) Problematic selection of phantom sources

(b) Sector-based neighbour selection in WRS

Figure 3.7: Angle-based privacy solutions

$\sum_{i=1}^{n} \frac{L_i}{p_i} p_i \approx nL$, where L is the mean length of all paths. However, this approach is only theoretically feasible. In practice, the generation of n truly parallel paths is a complex task, especially in large-scale sensor network deployments. It is also impractical for sensor nodes to store a large number of routing paths locally. Moreover, some of these paths may become useless over time due to the death of nodes or due to simple disruptions performed by an attacker in order to force the source node to use some particular paths. Finally, since the paths are parallel to each other, retrieving several packets from any of the paths provides a good idea of the direction to the source. This would significantly reduce the expected traceback time for the adversary.

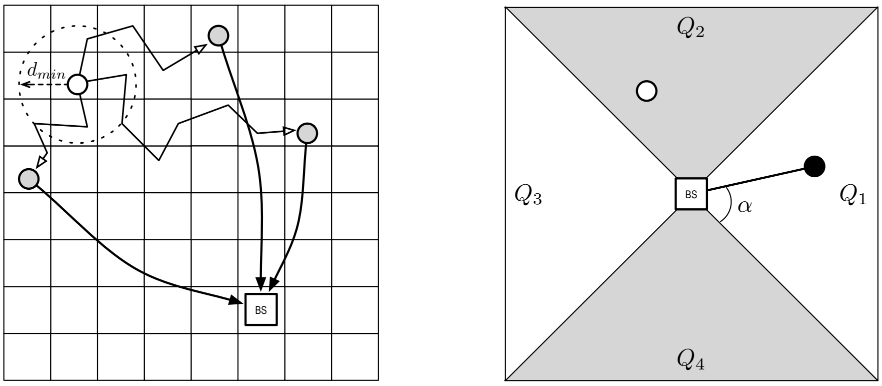
Besides developing the Random Parallel routing (see Section 3.2.1), Wang et al. [140] proposed the Weighted Random Stride (WRS). This algorithm is similar to PRLA in the sense that both of them make routing decisions probabilistically based on the inclination angle of its neighbours. Whenever a sensor node transmits a message to the base station it uses two parameters to guide the path, a forwarding angle and a stride. First, the data source randomly picks a forwarding angle and chooses a neighbour that matches that angle. After receiving the message, the node uses the same forwarding angle to select a new neighbour. This process continues until the stride, which defines the number of hops for a particular forwarding angle, reaches zero. Once the stride expires, the recipient node selects a new forwarding angle and starts a new stride[5]. In practice, instead of sensor nodes having to store the forwarding probabilities of all

[5] The stride is set to a value of 5 for a large scale sensor network with an average of 20 neighbours per node during the simulations

their neighbours, they are divided into closer and further nodes. Closer nodes are additionally divided into sectors and only nodes from these sectors are selected to forward the packet. In order to produce larger routing paths and thus deter the traceback attacks, sectors with larger inclination angles are prioritised. Within a particular sector, the node selects the neighbour which has the largest forwarding step. For example, in Figure 3.7b, sectors 1 and 6 are more likely to be chosen than sectors 2 and 5, and sectors 3 and 4 are the least likely. The main difference between this approach and PRLA is that in WRS there are no phantom sources from where the packets are finally routed to the base station using a single-path approach.

Li et al. [73, 110] proposed Routing through a Random selected Intermediate Node (RRIN) as another solution to the problem of selecting phantom sources close to the data source[6]. The authors assume that the network is divided into a grid and that each node knows its relative location (i.e., cell position) in the grid as well as the grid dimensions. In this way, instead of making each node in the walking phase take routing decisions independently, the source node can pick a random point in the field and send the packet to that location. The source node does not know whether there is a node in that particular location but in that case, the node closest to that location becomes the point from where the packet is finally transmitted to the base station using the shortest communication path. Li et al. propose two versions of RRIN. In the first version, the intermediate point is chosen uniformly at random but it is forced to be placed at least at a distance d_{min} from the source as shown in Figure 3.8a. The main drawback to this scheme is that the probability of being selected as an intermediate node is proportional to the distance to the data source. As a result, the intermediate nodes concentrate around the location of the source node and no mechanism prevents them from being picked from the proximities of the source-destination shortest path, which was one of the problems addressed by PRLA and WRS. In the second version of RRIN, any location in the network has the same probability of being selected as the random intermediate point. The consequence is that some intermediate nodes will be very close to the data source thus exposing its location while some others will be extremely far, not only resulting in energy-intensive paths but also

[6] They also argue that directed random walks leak information about the data source since the forwarding direction must be contained in the packet headers in order to allow nodes to route packets correctly. However, this information may be encrypted or encoded in the payload thereby alleviating the problem.



(a) Distance-based RRIN

(b) Quadrant-based multi-intermediate selection

Figure 3.8: Routing through Random selected Intermediate Node(s)

in more chances for the adversary to trace packets.

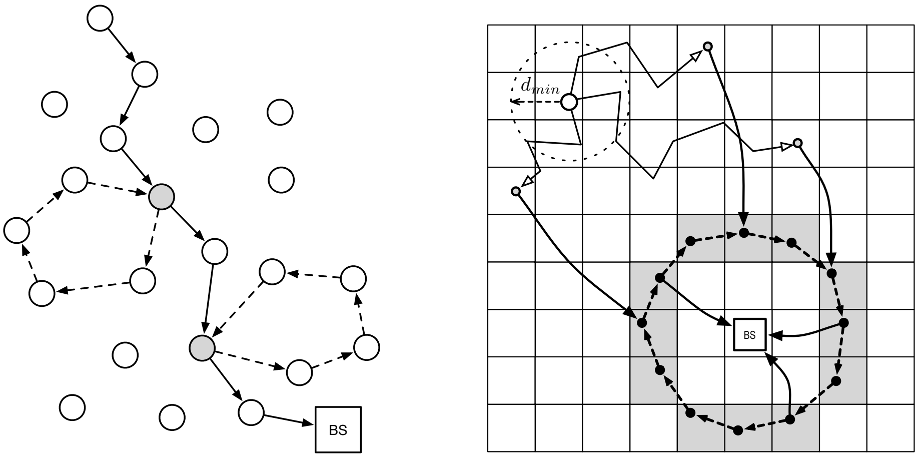
The RRIN scheme has been extended and used in several other research papers. In [72], Li and Ren propose two schemes that use multiple random intermediate nodes instead of a single one. In the angle-based multi-intermediate node selection, the source node selects a maximum angle β to limit the location of the last intermediate node within the range $(-\beta, \beta)$. Once the maximum angle has been determined, the source node uniformly chooses a random angle θ between itself and the node with respect to the base station, such that $\theta \in (-\beta, \beta)$. Then, the data source selects the rest of the n intermediate nodes to be evenly separated between itself and the final intermediate node. In the quadrant-based multi-intermediate node selection, each sensor node divides the network into four quadrants in such a way that it is placed in the first quadrant and the base station is in the middle. The source node location is determined within the first quadrant based on a random angle α. The last intermediate node is selected to be somewhere within its adjacent quadrants, namely quadrant 2 and 4 as shown in Figure 3.8b. Both extensions ensure that nodes are neither selected from behind the base station nor close to the shortest-path between the data source and the destination. However, it is not fully clear why it is necessary to use multiple intermediate nodes instead of a single intermediary.

The Sink Toroidal Routing (STaR) routing protocol [75, 76] is also designed to improve upon the initial RRIN designs. More precisely, it has been designed to reduce the energy cost associated with the selection of pure random intermediate nodes in the field. To that end, the source node picks random points within a toroidal region around the base station, which guarantees that intermediate nodes are, at most, a given distance from the destination but also not too close in order to prevent traceback attacks. The toroid is defined by three parameters: the centre of the toroid (x_0, y_0), where the base station is placed; r, the inner edge of the toroid; and R the outer edge of the toroid. Therefore, for each message a source node picks a distance value d uniformly from the interval $[r, R]$ and an angle θ from $[0, 2\pi]$. The intermediate node will be the one closest to the point $(x, y) = (x_0 + d\cos\theta, y_0 + d\sin\theta)$. The main drawback to this solution again has to do with the selection of problematic intermediate nodes not only between the source and the base station but also behind it.

Network Loop Methods

A completely different approach to deceive local adversaries consists of the creation of network loops. A network loop is basically a sequence of nodes that transmit messages in a cycle in order to keep the adversary away from the real direction towards the data source or to cover the presence of real traffic.

The Cyclic Entrapment Method (CEM) [98] is intended to set traps in the form of decoy messages to attract the adversary and distract him from the true path to the data source for as long as possible. After the deployment of the network, each sensor node decides whether it will generate a network loop with a given probability. Then, the node selects two neighbouring nodes and sends a loop-creation message that travels h hops from the first to the other neighbour. All the nodes receiving this message become loop members. During the normal operation of the network, a loop is activated whenever a loop member (i.e., activation node) receives a real packet being routed from a source node to the base station. Interestingly, CEM is not a routing protocol itself but rather an add-on that can be used with different routing protocols to enhance source-location privacy. This implies that, when used in conjunction with single-path routing, real traffic reaches the base station in the shortest time possible without incurring extra delays. Figure 3.9a depicts such a scenario where two loop members (in grey) become activation nodes after receiving real traffic. During a traceback attack, when the adversary reaches an activation node he must decide which packet to follow. If he chooses the fake message he is trapped in the loop for h hops until he realises. However, an skilled adversary might avoid loops by observing the angle



(a) Cyclic Entrapment Method (b) Network Mixing Ring

Figure 3.9: Network Loops Methods

of arrival of packets since those with a larger inclination angle are more likely to lead to a loop.

The information Hiding in Distributed Environments (iHIDE) scheme by Kazatzopoulos et al. [63, 64] is another solution that uses network loops. In this scheme, the sensor network consists of a set of ring nodes which are interconnected with each other and with the base station by means of a network bus. This arrangement is similar to the one depicted in Figure 3.9a but in iHIDE all sensor nodes are either bus or ring nodes. During the data transmission period, a source node that wishes to communicate data to the sink first sends the data to the next ring member in a (counter-)clockwise direction[7]. When the bus node receives the packet, it forwards it to the next bus node closer to the sink but the packet continues to loop in the ring for a random number of hops. As the packet travels through the bus, each bus node decides, based on a given probability, to forward the packet into its own ring or to directly submit it to the next bus node. The main limitation to iHIDE is that because it has such a well defined architecture and roles for the nodes it is easy to learn the topology of the network and thereby identify the bus and the rings. Once a bus node has been reached, the adversary can wait until he observes that the bus node receives a message from another bus node that it forwards to the next one. This implies that somewhere in a previous ring there is a data source. In this way, the adversary can slowly

[7] In the case that the sensor node belongs to multiple rings simultaneously it randomly selects one of them to forward the message.

reduce his uncertainty.

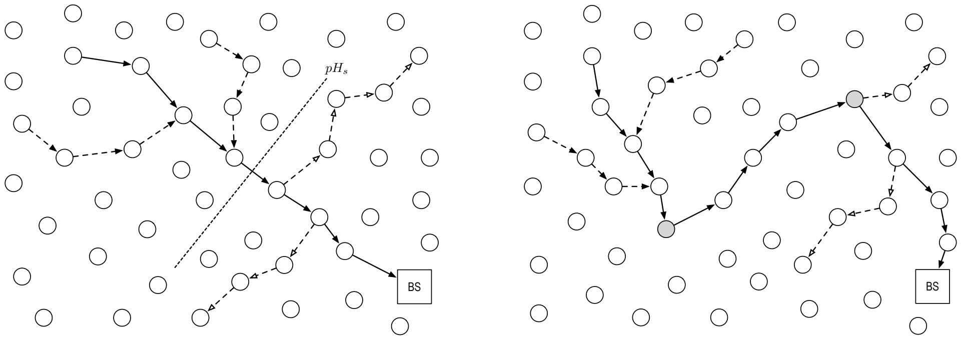
The Network Mixing Ring (NMR) scheme [71, 74] creates a virtual ring of nodes surrounding the base station whose aim is not to trap the adversary but to mix up real messages with fake traffic in order to make them indistinguishable to the adversary. This scheme consists of two phases. In the first phase, the source nodes picks a random intermediate node which is in charge of initiating the next phase. The selection process is based on the distance-based RRIN approach described in Section 3.2.1. In the second phase, the intermediate node sends the packet to the closest node in the network mixing ring. Once there, the packet is relayed clockwise for a random number of hops before being finally submitted to the base station. Within the mixing ring there are a few nodes that generate network traffic, namely vehicle messages. These messages carry several data units, which are all initially filled with garbage but as real messages enter the ring the fake data units are replaced. To further complicate traffic analysis, vehicle messages are re-encrypted at every hop. In this way, even if the adversary reaches the ring node that forwarded the data to the base station he is unable to figure out the entry point of that packet to the ring. Moreover, entry points are changed over time. The whole process is depicted in Figure 3.9b, where the grey cells represent the area defining the network mixing ring. A major limitation to this approach is the increased energy consumption at the ring nodes, which are more likely to deplete their batteries than other nodes. This event not only ruins the source protection mechanism but also isolates the sink from the rest of the network, rendering the whole system useless.

To diminish the energy imbalance between ordinary sensor nodes and ring nodes, the authors propose predefining several rings and activating only one at a time according to the residual energy of their members [71]. Additionally, they briefly discuss the possibility of having several active rings simultaneously to improve the level of protection of the data sources. More recently, Yao et al. [154] have continued with the idea of organising the network using a multi-ring approach to protect source-location privacy. This scheme consists of three phases: initialisation, path diversification, and fake packet injection. During initialisation, the base station floods the network with a discovery message which includes a hop count. This process allows sensor nodes to obtain their distance to the base station as well as to determine which of their neighbours are at the same distance, which means that they belong to the same ring. In the following phase, the data source picks, uniformly at random, two rings (one closer and one farther) and an angle α between zero and π. Then, the data packet is sent out to the farther ring and once there it is relayed counterclockwise until the angle is reached. From this point, the packet is sent to the closer ring and once more travels counterclockwise for an angle $\beta = \pi - \alpha$. Finally, the packet is routed directly to the base station. During transmission of real traffic on the rings, fake packets are injected by the nodes on contiguous rings to further complicate traffic analysis. Clearly, these ring-based solutions require the network to be densely populated in order to enable the creation of full rings.

Fake Data Sources

The idea of using fake data sources was first suggested by Ozturk et al. [100]. They proposed two strategies, namely Short-lived and Persistent Fake Source, to simulate the presence of real events in the field by making some sensor nodes to behave as true data sources. In the first strategy, whenever a sensor node receives a real message it decides, based on a particular probability distribution, whether to generate a fake message and flood the network with it. This scheme provides a poor privacy protection since fake data sources are ephemeral. The second strategy aims to prevent this by creating persistent sources of fake messages. Each sensor node decides with a probability to become a fake data source. The efficiency of this strategy is very much dependent on the positioning of the fake data source. If fake data sources are far from a real data source it helps to improve the safety period significantly, otherwise it may lead the adversary to the real data source.

Chen and Lou [29] designed several solutions to protect location privacy based on the use of fake messages, namely the Bidirectional Tree (BT) scheme, the Dynamic Bidirectional Tree (DBT) scheme, and the Zigzag Bidirectional Tree (ZBT) scheme. These solutions are intended to protect both source- and receiver-location privacy simultaneously but we cover them here in full detail to avoid the duplication of contents across different sections. In the BT scheme, real messages travel along the shortest path from the source to the sink and several branches of fake messages flow into and out of the path. To that end, before the transmission of data messages, the source node sends a packet containing its own hop count H_s along the shortest path. Those nodes in the path whose distance to the sink is greater than $(1-p)H_s$, being p a network-wide parameter, will generate an input



(a) Bidirectional Tree Scheme (b) Zigzag Bidirectional Tree Scheme

Figure 3.10: Bidirectional Tree Schemes

branch with a given probability[8]. Similarly, the nodes satisfying pH_s will choose whether to generate an output branch. This solution is depicted in Figure 3.10a, where dashed arrows represent (input or output) fake branches. The idea behind the creation of fake branches is to misdirect the adversary from the real path while event data reaches the base station in the shortest time possible. However, it is not difficult for a skilled adversary to realise that nodes deviating from the already travelled path are fake branches.

To prevent the adversary from easily obtaining directional information, the DBT scheme suggests that real messages should travel with a forward random walk (see FRW in Section 3.2.1) instead of using a single-path routing approach. When a node receives a real message it decides the next hop uniformly at random from its list of those neighbours closest to the base station. Similar to the BT scheme, fake branches are created in order to complicate packet tracing attacks further. In this case, input branches are generated with a probability when the hop count is smaller than $H_s/2$, and output branches otherwise. The ZBT is another scheme that has also been devised to prevent leaking direction information. To that end, real packets zigzag along three segments: from the source node to a source proxy, from there to a sink proxy, and finally to the real sink. First, two candidate sink proxies are selected, one on each side of the sink and at a distance of h hops. Then the sink and the two proxies initiate a flooding so that each node learns its distance from each of them. In this way, the source node

[8] Input messages cannot originate from a node belonging to the shortest path but from a remote node. The authors do not specify how remote sources of fake data are selected. A possible solution is to send a message on a directed random walk from the node in the shortest path.

can select the sink proxy which is furtherest away from itself. Having a single sink proxy may imply that a source node is very close to that proxy, which would negatively impact source-location privacy. Before the transmission of data to the sink, the source node picks a source proxy h hops away from itself. The source proxy should be selected in such a way that it is not close to the sink. However, this is not a trivial task unless the nodes are aware of the physical location of all other nodes. Finally, during the data transmission phase, each node in the path generates fake branches with a given probability. In the segment from the source node to the source proxy, the fake packets flow into the path, and in the segment from the sink proxy to the sink, the packets flow out. No branches are generated in the segment connecting the source and sink proxies. The operation of the ZBT scheme is depicted in Figure 3.10b, where grey nodes represent the source and sink proxy nodes. This scheme presents the same limitation as the original BT scheme, that is, fake branches can be eventually discarded. Either the attacker discards a fake branch after tracing it or due to a unusual inclination angle. Moreover, the segment between the source and sink proxies does not generate any branches, which implies that an attacker can easily determine their locations and from there reach its target.

Jhumka et al. [56] developed two solutions, namely fake source (FS) 1 and 2, to investigate the effectiveness of using fake data sources to protect source-location privacy. Both solutions are built on top of a baseline flooding protocol. In FS1, the data source floods the network with a data message containing the event data and a hop count. When this packet reaches the base station, it generates an away message containing the distance between itself and the data source, and floods the network with it. The away message is intended to reach all nodes at the same distance as the source to the sink and make them transmit a choose message. This new message is forwarded to nodes further away, which decide to forward it based on a given probability. When the hop count of the choose message reaches 0, it generates a random number and, if above a given threshold, the node becomes a fake data source. The FS2 protocol is very similar to FS1, the difference is that in FS2 all the nodes that receive a message forward it, while in FS1 the forwarding of messages is determined by a given probability. Consequently, more nodes are likely to become fake data sources in FS2 and thereby the level of protection achieved by this scheme is better at the expense of increased energy consumption.

3.2.2 Global Adversaries

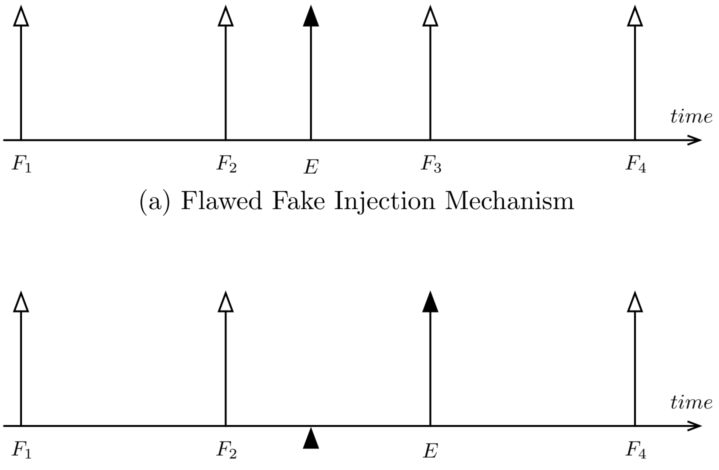
The aforementioned techniques are only effective against adversaries performing traceback attacks with a limited hearing range. A more powerful adversary is capable of monitoring the behaviour of a larger number of nodes simultaneously, which allows him to better correlate messages and guess routing paths. In particular, global adversaries are capable of monitoring all the traffic generated and forwarded in the network. Such adversaries can easily detect the data sources among mere intermediaries because sensor nodes are programmed to report event data to the base station as soon as it is detected.

Dealing with global adversaries is very challenging especially in scenarios where there exist topological, functional or hardware constraints, as we learnt in Chapter 2. There are two main approaches to hide the location of data sources, either using fake traffic to cover the presence of event messages or introducing significant delays in the transmission of messages. Both solutions present some disadvantages in the sensor domain. The former implies a massive energy waste while the latter has a negative impact on the ability of the network to provide the base station with timely reports about events, which is essential for time critical applications.

Most solutions in this area have concentrated on the injection of fake traffic to provide event source unobservability and a huge research effort has been devoted to making these solutions as energy-efficient as possible.

Dummy Traffic Injection

The threat of global adversaries was first considered by Mehta et al. in [89], where they proposed the Periodic Collection scheme. This scheme makes every node transmit fake messages at regular intervals to hide the presence of events in the field. However, it is not as simple as sending fake messages at a constant rate because the occurrence of an event message would change the message transmission pattern as shown in Figure 3.11a. This figure depicts a timeline where the transmissions of real and fake packets are represented by arrows with white or black heads, respectively. In the Periodic Collection scheme, sensor nodes transmit messages at a given rate $\mathcal{R}$ regardless of the presence of events. Instead of transmitting a message immediately after the detection of an event, the message is temporarily stored until the next scheduled transmission time, as shown in Figure 3.11b. Since real and bogus traffic are indistinguishable from each other, this



(a) Flawed Fake Injection Mechanism

(b) Perfect Event Source Unobservability

Figure 3.11: Periodic Fake Packet Injection

method provides perfect event source unobservability because the transmission rate is not altered by the presence of events.

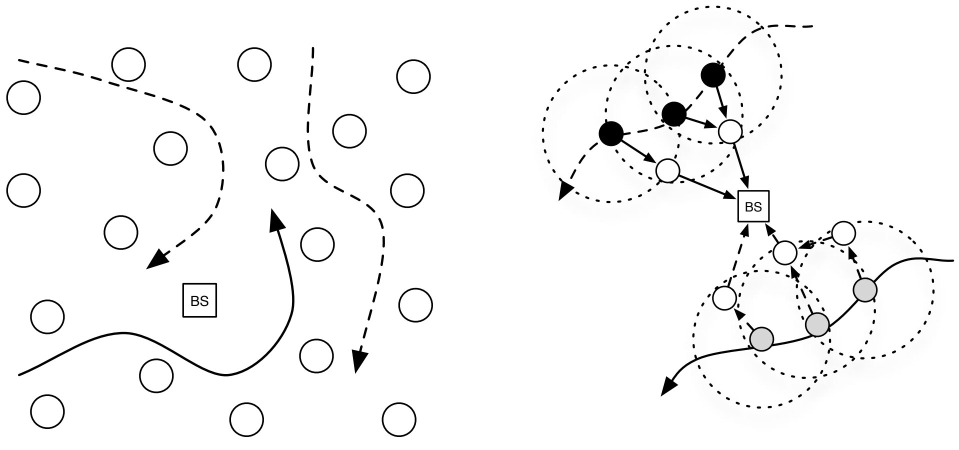
As event messages need to be delayed until the next scheduled transmission time, this poses a serious limitation in time-critical applications. Intuitively, the delivery delay can be reduced by changing scheduling in order to have shorter inter-transmission times. However, this impacts negatively on the energy waste of the network. Therefore, the transmission rate must be carefully adjusted in order to ensure the durability of the network without incurring an excessive delay in the delivery of messages to the base station.

Energy-Aware Approaches

There has been an extensive body of research which focuses on reducing the overhead imposed by the injection of fake messages at regular intervals by all sensor nodes. These proposed solutions have approached the problem in different ways: simulating the presence of events in the field, filtering out fake traffic, using already existing traffic to convey event data, and sending messages according to a given probability distribution.

A first attempt to reduce the overhead produced by the Periodic Collection scheme was devised by Mehta et al. [89]. This scheme, called Source Simulation is based on the idea of saving energy by reducing the number of nodes transmitting fake messages. Instead of making all nodes send out messages at regular intervals, the network simulates the presence of real events in the field. The main problem with this approach lies in the difficulty of accurately modelling the movement of an object so it appears as real to the adversary. In such a case, having a static subset of sensor nodes transmitting fake messages is not enough to deceive an attacker. Therefore, sensor nodes must be carefully programmed to transmit fake messages following a coherent pattern that resembles a real object. Moreover, this process should be carefully tailored to any type of asset being monitored, which turns it into a challenging and laborious protection mechanism. Mehta et al. propose a source simulation protocol as follows. During network deployment, a set of L nodes are preloaded, each with a different token. These nodes generate fake traffic during the data transmission phase and after a predefined period of time, the token is passed to one of its neighbours (possibly itself) depending on the behaviour of real objects. The size of L determines the level of protection as well as the energy consumed by the network.

The Unobservable Handoff Trajectory (UHT) presented by Ortolani et al. [97] is another solution that simulates the movements of objets in the field to preserve source-location privacy against global adversaries. This solution focuses on the protection of events originating at the perimeter of the network and eventually expiring at some point inside it (see Figure 3.12a). A clear example is the transportation of goods to an industrial area. The UHT is a decentralised and self-adaptive scheme that generates fake mobile events with the same probability distribution as real events. Real events follow a Poisson distribution of rate l while fake events are generated with rate $k-l$. In consequence, the overall distribution of messages in the network follows a Poisson distribution of ratio k thus covering real events. The generation of dummy events starts at the perimeter of the network and propagates for a number of hops according to the length of real events. Each perimeter node decides to generate a new dummy event independently based on a Poisson with parameter $k - l/P$, where P is the number of perimeter nodes and l, although unknown, can be estimated by choosing a statistical estimator. To do this, perimeter nodes record the number of real events they observe over a time window. The propagation of fake event messages works as follows. All nodes within the radius of a fake node receive the fake packets sent towards the base station. This packet contains who will be the next fake source in the path and also the length of the current event. This process is represented in Figure 3.12b, where fake sources are shaded in grey and real sources in black while fake and real messages are represented with dashed and ordinary arrows,



(a) Real and fake mobile events (b) Modelling of event propagation

Figure 3.12: Unobservable Handoff Trajectory

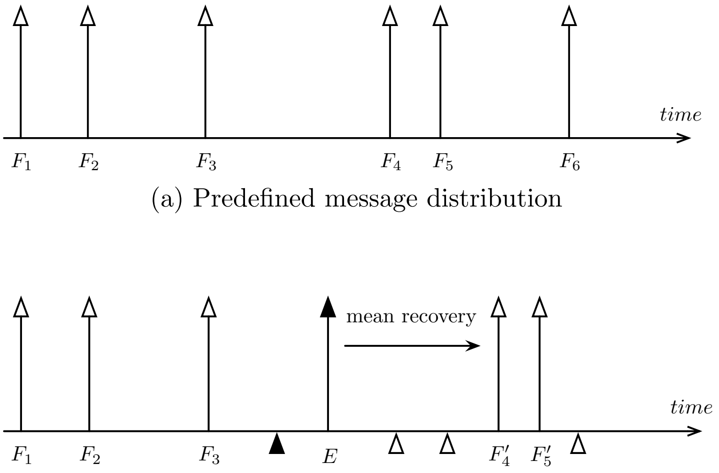
respectively.

Besides the cross-layer scheme described in Section 3.2.1, Shao et al. [128] proposed another version of the same solution that can be useful in the protection of source-location privacy against global adversaries. This alternative protocol is very similar to the Periodic Collection proposed by Mehta et al. but the main difference is that instead of using ordinary network traffic it takes advantage of the beaconing phase. This scheme also provides perfect event source unobservability at no additional cost since event data is hidden within beacon frames, which are periodically broadcast regardless of the occurrence or not, of events in the field. However, since the time between consecutive beacons is relatively large, the solution is only practical for some applications where no tight time restrictions exist. Also, this solution is inadequate for large-scale sensor networks since the delivery time is highly dependent on the distance from the data source to the base station.

In order to reduce network traffic while maintaining source unobservability, Yang et al. [149] proposed a bogus traffic filtering scheme. In this approach, the network is divided into cells and some sensor nodes operate as filtering proxies. Cells send real or fake messages at a given rate and on their way to the base station they reach some of these proxy nodes. Upon the reception of traffic, a proxy node discards bogus traffic and real traffic is temporarily buffered and re-encrypted before being forwarded[9]. In the case there are no event messages available, a proxy node sends encrypted dummy messages to prevent the attacker from learning which proxies are receiving real traffic from some of its associated cells. Two filtering schemes are proposed, the Proxy-based Filtering Scheme (PFS) and the Tree-based Filtering Scheme (TFS). The PFS is the baseline approach where a number of nodes are selected as proxies but the traffic generated by each cell is only filtered once by its default proxy node. In TFS, a multi-layered proxy architecture is proposed to further reduce dummy traffic. As packets move towards the base station they can be processed by several proxy nodes, which reduces fake traffic at the expense of increased network delay due to the buffering at each proxy node. Thus, the number and location of proxy nodes is very important to the performance of the solution. It should be noted that a drawback to this solution is that an attacker can still use rate monitoring techniques to identify the proxy nodes, which are important for the operation of the network.

Another branch of research has concentrated on the concept of statistically strong source unobservability to reduce message delivery time and increase the lifetime of the network. This concept was introduced by Shao et al. [129] to relax the tight requirements of perfect event source unobservability while maintaining a statistical assurance on the protection of data source. Before deployment, sensor nodes are configured to transmit according to a message distribution F_i as depicted in Figure 3.13. During the data transmission phase, when an event E occurs, the real message can be transmitted before the next scheduled transmission, F_4, without altering the parameters (e.g., the mean and variance) of the distribution. This process is depicted in Figure 3.13b. Sensor nodes keep a sliding window of previous inter-message delays $\{\delta_1, \delta_2, ..., \delta_{n-1}\}$ and, upon the occurrence of an event, δ_n is set to a value very close to 0 and gradually incremented by a small random number until the whole sliding window passes an Anderson-Darling goodness of fit test. Thus, the real event transmission can be sent ahead of the scheduled time without alerting the adversary even if he performs statistical tests on inter-message delays. The solution proposed by Shao et al. includes a mean recovery mechanism which delays subsequent transmissions because the presence of bursts of real messages might skew the mean of the distribution.

[9] Cells are assumed to share pairwise keys with proxy nodes to allow them distinguish real from fake messages.



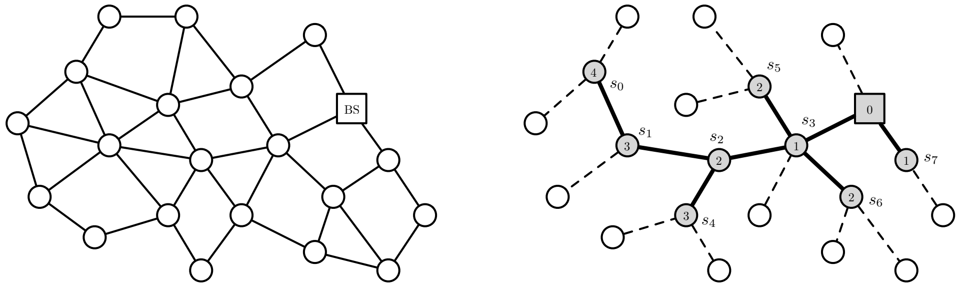
(a) Predefined message distribution

(b) Message distribution adjustment

Figure 3.13: Statistically strong source unobservability

Recently, Alomair et al. [7, 8] showed that a global adversary has more efficient ways of breaking statistically strong unobservability. Instead of focusing on the inter-message delays of a single sliding window, the attacker might try to spot differences between any two sliding windows in order to detect the presence of real events. Therefore, the strategy of the adversary to distinguish between an interval (i.e., a sliding window) containing real events from another one with no real events, is to identify short inter-message delays followed by long inter-message delays. These patterns are common in intervals containing real events because the delay of real messages is usually shorter than the mean in order to reduce the latency, and subsequent messages are delayed in order to adjust the mean of the distribution as proposed by Shao et al. [129]. To the contrary, inter-message delays are independent identically distributed random variables in fake intervals. Consequently, by counting the number of short-long inter-message delays an attacker might be able to distinguish intervals containing real events. The solution proposed by Alomair et al. to reduce the success probability of the attacker is to make fake intervals resemble intervals with real events by introducing some statistical interdependence between fake inter-message delays.

Proano and Lazos [105] pointed out that the adversary cannot exactly determine the transmission rate of each and every sensor node. This is due to the fact that a global vision of the network is usually achieved by means of an adversarial sensor network. Each adversarial node only knows the number of packets sent within its hearing range but it is unaware of which node is sending each of the



(a) A graph representation of the WSN

(b) A potential MCDS

Figure 3.14: Minimum Connected Dominating Set

packets unless these data are present in the packet headers. As a result, not all sensor nodes need to be active sources of fake traffic to deceive the adversary. The problem of reducing the number of fake data sources is solved by partitioning the network into a minimum connected dominating set (MCDS) rooted at the base station. The MCDS covers the whole network by using the minimum number of nodes in such a way that each node in the network either belongs to the MCDS or is one hop away from it, as depicted in Figure 3.14. In this way, the nodes in the MCDS transmit (real or fake) traffic at a given rate $\mathcal{Z}$ and the rest of the nodes regulate their transmissions in order to conform to the statistical traffic properties observed by an eavesdropper. Since the location of the eavesdropper is unknown, each sensor node divides its hearing range into several regions so as to consider all potential locations and computes a rate that satisfies the original rate $\mathcal{Z}$ for all of them. Later, in [106], the same authors added a deterministic assignment scheme for coordinating sensor transmissions and thus reduce end-to-end delay for real packets. Time is divided into intervals of duration T and each interval is in turn divided into subintervals of duration $\frac{T}{l}$, where l is the height of the MCDS. Nodes deeper in the MCDS are scheduled to transmit sooner, so that any real packet reaches the sink at the end of each interval. For example, in Figure 3.14b, each time interval I_k is divided into four subintervals since the maximum depth of the MCDS is four. Sensor node s_0 transmits at subinterval I_k^1, node s_1 at subinterval I_k^2, and so on.

Previous solutions have considered a passive global attacker in the sense that he does not check in the field whether his observations lead to an actual data source. Yang et al. [150] consider a global attacker who, upon detecting suspicious cells devises an optimal route to efficiently visit these spots. As usual, the attacker performs traffic analysis on the network by deploying an adversarial network and based on his observations he obtains a suspicion level for each cell. Then, he defines a suspicion threshold to determine which cells to visit and in what order. Since this problem has a factorial time complexity on the number of suspicion cells (i.e., $\mathcal{O}(s \cdot s!)$), Yang et al. propose two potential strategies to find a (pseudo-)optimal route to visit all suspicious cells. The first strategy is based on a greedy algorithm, which ends in polynomial time but is not globally optimal, and the second one is a dynamic programming algorithm, which finds the optimal solution but requires an exponential time to finish. Subsequently, the authors evaluate the impact of the proposed attacker model to two existing solutions: statistically strong source unobservability and source simulation. They conclude that the former behaves well when the rate of real messages to be delivered is low while the latter approach is suitable when the rate is high. As a result, Yang et al. propose a dynamic approach that combines the merits of both solutions by switching from the one to the other based on the load of the network.

3.2.3 Internal Adversaries

Some adversaries might be able to compromise and control a subset of nodes from the legitimate network. These nodes become internal adversaries since they can participate in the same tasks performed by any other network node. Thus, internal adversaries can provide the attacker with any information contained in the packets they forward since they share cryptographic material with their neighbours.

The solutions devised to deal with these types of attackers are very limited and their approaches rather diverse. To the best of our knowledge, so far there are only three solutions and they have concentrated on the implementation of a trust-based routing solution, the modification of packets in transit, and the decoupling of the location where the data is sensed, from the location where it is temporarily stored before it is collected by the base station. Next we review them in more detail.

The Identity, Route and Location privacy (IRL) algorithm is presented by Shaikh et al. [124] as a network-level privacy solution. The primary goal of this solution is to provide source anonymity and location privacy as well as provide assurance that packets reach their destination. Although the authors do not consider the threat of internal adversaries, one of its features is suitable for

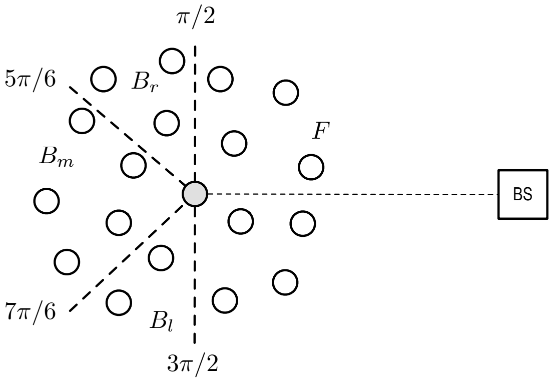


Figure 3.15: Neighbours partition in IRL

just this purpose. The authors introduce the notion of trust and reputation to prevent routing through misbehaving adversaries. First, each node classifies its neighbours into four groups depending on their position with respect to the base station: forward (F), right backward (B_r), left backward (B_l), and middle backward (B_m), as shown in Figure 3.15. Furthermore, each node classifies its neighbours as either trustworthy or untrustworthy based on the number of successfully forwarded packets. Nonetheless, the calculation of the trust values could be extended to incorporate new parameters, such as the presence of communications with external entities or with other non-neighbouring nodes in order to identify internal adversaries. When a node needs to send a message to the base station, it checks whether there are any trustworthy nodes it can select in the direction of the base station. From among all the trustworthy nodes it picks one uniformly at random. If there are no trustworthy nodes, the same process is repeated for B_r and B_l. As a last resort, the node tries to send the packet in the opposite direction to the base station. In the case no trustworthy nodes are found, the node simply drops the packet. Therefore, each message follows a different (random) path composed of trustworthy nodes only.

Additionally, IRL includes a renaming mechanism to protect the identity of the data source. Whenever a node receives a packet it replaces the identifier contained in its header with its own before forwarding it. In this way, dishonest en-route nodes are unable to determine whether the sender is the real data source or a mere intermediary. This implies that the identity of the real data source is conveyed in the packet payload, encrypted with a pairwise secret shared with the base station. The use of end-to-end encryption is efficient against internal adversaries but it impedes the use of data-aggregation mechanisms, which is a useful feature for reducing network traffic and thus preserving energy.

Pongaliur and Xiao [103, 104] propose a more sophisticated packet transformation scheme called Source Privacy under Eavesdropping and Node compromise Attacks (SPENA) based on the application of some cryptographic operations on the packets at dynamically selected nodes in the route to the base station. Packets have the following structure: $\{DstID$, $SrcID\ Hash$, $Obfuscating\ Partial\ Hash$, $Rehash\ Seed$, $Payload\ Length$, $Payload \mid SrcID$, $Filler\}$, and nodes are preloaded with two unique hash functions, a mapping function fp that returns 1 with probability p and 0 with probability $1-p$, a rehash function, and a symmetric key shared with the base station. One of the hash functions H_i is used to generate a hash chain $(h_i^1, h_i^2, \ldots, h_i^n)$ used in reverse order as the identities of the nodes and the other hash function F_i is used in conjunction with the mapping function to determine whether a node should modify the packet or not. In particular, a node j transforms a received packet if $f_p(F_j(Rehash\ Seed)) = 1$. Nodes that transform packets in transit are called rehashing nodes. At the data source, the $SrcID\ Hash$ field is loaded with an element of its hash chain (i.e., h_i^m) and later replaced by a rehashing node j by a value of its own hash chain (i.e., h_j^k). The $Obfuscating\ Partial\ Hash$ (OPH) is initially set to the next element of the hash chain concatenated with the payload and encrypted with its symmetric key (i.e., $E_{K_i}(h_i^{m+1}|Payload)$). A rehashing node j generates a new OPH_j by first applying the rehashing function to the received OPH and then encrypts it with its own key. Additionally, the rehashing node concatenates the $SrcIDHash$ obtained from the received packet to the payload, which is then encrypted with its own symmetric key. It also uploads the new payload length and subtracts that amount of bits from the filler to keep the packet size unchanged. At the base station, the payload is recursively decrypted until the $SrcID\ Hash$ of the true data source is found. Finally, the base station checks the validity of the OPH. The verification process requires the base station to keep track of the hash chains of all the nodes in order to find the key corresponding to each of concatenated the hash values. Another limitation to this approach is that the attacker can trivially learn the real size of the payload by inspecting its corresponding header field and thereby guess the number of modifications the packet has suffered based on the probability p of the mapping function. By having access to this information the attacker can estimate its distance to the data source.

The last solution is called pDCS [130] and its aim is to provide security and privacy in Data-Centric Sensor (DCS) networks with an itinerant base station. In DCS networks, there are two types of nodes: sensing nodes, which collect and forward information about events of interest, and storage nodes, which temporarily store the data from a subset of sensing nodes and respond to the queries of the itinerant base station. The relationship between sensing and storage nodes is defined by a publicly known mapping function that determines where the data is stored. In this way, the data can be accessed more efficiently but it also allows an attacker to easily determine which nodes to compromise if he is interested in a particular type of data. After compromising such nodes, he can also identify the location where the data was originally collected. *p*DCS is intended to protect against this type of threat. In particular, it concentrates on preventing node compromise and mapping attacks, that is, impeding the retrieval of any event data stored in storage nodes as well as preventing the attacker from identifying the relationship between sensing and storage nodes. The proposed scheme is based on the use of a secure mapping function[10] and the storage of encrypted data in a remote location. In the case the adversary compromises a storage node he is not able to decrypt the data contained in it because these data are encrypted with the key of the sensing nodes which collected them. If a sensing node is compromised, the attacker cannot determine where previous data was stored because the secure mapping function prevents this from happening. Moreover, when a node is found to be compromised there is a node revocation mechanism in order to prevent the attacker from obtaining the location of future event data. Finally, the authors suggest protecting the flow of data from the sensing to the storage node by means of any existing source-location privacy solution.

3.3 Receiver Protection

Receiver-location privacy refers to the protection of the destination of messages but it primarily concentrates on hiding the location of the base station. This device demands exceptional protection measures given its importance for both the physical protection of the network and strategic reasons. An attacker aware of the location of the base station may compromise it for his own benefit. For example, the attacker may be interested in gaining access to the data collected by

[10] A secure mapping function is basically a keyed hash function that uses as input the type of event and other secret information shared by a group of nodes.

the network, change configuration and operation parameters, or even destroy the base station and thereby render the whole system useless. Additionally, the base station provides strategic information because it is usually housed in a relevant facility (recall the scenario depicted in Section 1.4.1).

The location of the base station is exposed due to the peculiar communication pattern of WSNs. Each sensor node transmits data messages to a single base station using a multi-hop routing protocol, which results in a high volume of traffic in the proximities of the sink. Intuitively, the solution is to normalise the traffic load by making each sensor node transmit, on average, the same number of messages. Thus, a baseline flooding protocol provides the maximum protection but it also incurs a prohibitive network overhead. Solutions in this area have concentrated on providing a sufficient protection level at a reasonable cost.

In the following we review the existing solutions according to the capabilities of the adversary. We analyse proposals dealing with local adversaries followed by solutions considering the threat of global adversaries. There are no solutions in the literature that study the threat of internal adversaries or node compromise attacks. To the best of our knowledge, the first receiver-location privacy solution to consider this type of threat has been developed as part of this dissertation and is presented in Chapter 5.

3.3.1 Local Adversaries

In a local adversarial model, the attacker usually starts at a random position in the network[11] and moves around until he overhears some transmissions in the area surrounding him. The typical types of attacks performed by an adversary who wishes to find the sink are: content analysis, rate monitoring, and time correlation. In content analysis, the adversary looks for any valuable information that might lead him to the base station in either the packet's headers or the payload. This attack may be taken a step further by adding undetectable marks to data packets as proposed by Shakshuki et al. [126] in order to allow the adversary to track them on their way to the sink. Nonetheless, content analysis is usually a poor source of information.

Additionally, an attacker can observe the packet sending times of neighbouring nodes in order to determine the direction of the communication flow. Assuming that the network is using a single-path routing protocol, the attacker can learn

[11] Placing the adversary at the edge of the network is, in our opinion, more realistic.

that a sensor node is closer to the base station than one of its neighbours if it is used as a relay. In other words, if a node transmits immediately after one of its neighbours, the former node is closer to the sink. Finally, in a rate monitoring attack, the strategy of the adversary is to move in the direction of those nodes with higher transmission rates since nodes in the vicinity of the base station receive more packets than remote nodes.

Next we analyse some basic countermeasures against the aforementioned attacks followed by a set of more advanced solutions which provide enhanced security to the base station. Most of these solutions aim to balance the amount of traffic between all network nodes by selecting the next hop based on some probability while other solutions attempt to disguise or emulate the presence of the base station at different locations. Again, some solutions may fall into several categories depending on the features analysed.

Basic Countermeasures

In order to prevent the aforementioned traffic analysis attacks, some basic countermeasures have been proposed. First, content analysis can be hindered by applying secure data encryption on a hop-by-hop basis. Deng et al. [39] suggest this process should be applied throughout the whole lifetime of the network but it is not easy to satisfy this requirement until each node shares pairwise keys with all its neighbours. Thus, they propose an ID confusion technique to conceal the source and destination during the route discovery phase. This technique is based on reversible hash functions so that when a node x sends a message to node y, it randomly selects an element from $C_x = \{h_x : x = H(x)\}$ as the source address, and an element from $C_y = \{h_y : y = H(y)\}$ as the destination address. Finally, it encrypts the whole packet with a network-wide shared key pre-loaded on all sensor nodes. A receiving node decrypts the message and, by reverting the hash function, it obtains the true sender and intended recipient.

During data transmission, sensor nodes must ensure that packets change their appearance as they move towards the base station. Each node in the path must decrypt any received packet and then re-encrypt it with the key shared with the next node in the route. However, even if the attacker cannot observe the contents of the packets, he can learn some information from packet sending times and eventually infer the relationship between parent and child (i.e., closer and further) nodes. To prevent this, Deng et al. [38, 40] propose applying random delays to the transmission of packets. Additionally, the authors suggest creating a uniform sending rate to prevent rate monitoring attacks. This can be achieved by making a parent node accept packets from a child node only if its own packet has been forwarded. In the case the parent node has nothing new to send, it can simply continue to send the same packet or inject dummy traffic.

There are some limitations to these basic countermeasures that require the development of further solutions. These limitations are related to the delay introduced at each forwarding node and the energy wasted due to the application of uniform data transmission rates. The following solutions aim to reduce these limitations.

Biased Random Walks

This category brings together solutions where the routing process is random but somehow biased towards the base station. The first solution we analyse here is also presented by Deng et al. [38, 40] and is called Multi-Parent Routing (MPR). The MPR consists of making each sensor node pick the next element in the path uniformly at random from its set of parent nodes. See in Figure 3.16 a comparison between a single-path routing and a MPR scheme. In Figure 3.16a all transmissions use the same transmission path, which is represented by a straight arrow, while in Figure 3.16b the paths followed by two different packets are represented. The MPR scheme obtains a better load balance as data packets spread within a band of nodes next to the shortest path from the data source to the base station. However, the traffic flow still points to the base station as the next communication hop is selected from the list of parent nodes.

To further diversify routing paths and introduce packets in different directions, the authors suggest combining MPR with a random walk (RW) routing scheme. In this version of the protocol, nodes forward packets to a parent node with probability p_r and to a randomly chosen neighbour with probability $1 - p_r$. Consequently, packets may not only travel towards the base station but in any other direction. In Figure 3.16c we depict two routing paths which at some points even move in the opposite direction to the base station. This scheme provides better security at the cost of a higher message delivery delay.

Similarly, Jian et al. [57, 58] propose in Location Privacy Routing (LPR) to make every sensor node divide its neighbours into two groups. The first group contains nodes which are closer to the base station and the second group contains

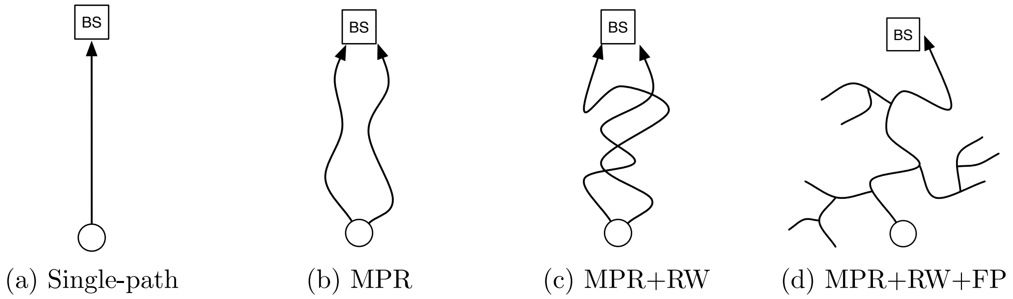


Figure 3.16: Schematic of Several Multi-Parent Routing Techniques

the rest of their neighbours. So, nodes forward packets to further nodes with probability P_f and to closer nodes with probability $1-P_f$. To ensure that packets reach the base station the value of P_f must be below 1/2. This implies that after a sufficient number of observations, the attacker is able to determine which of the neighbours of a node belong to each group. By following this strategy at different nodes, the attacker is able to infer the direction toward the data sink. To prevent this, the authors propose injecting fake packets in the opposite direction to the base station. When a node forwards a real packet, it generates with probability P_{fake} a fake packet to a random node in the group of further nodes. This packet travels for $M_f \geq 2$ hops away from the base station[12]. In general, the adversary cannot distinguish real from fake traffic which makes this solution secure since packets flow in any direction with an even probability. However, if the adversary observes a node that does not forward a packet he knows that it is a fake packet. As fake packets are sent to further neighbours exclusively, the adversary learns that the base station is in the opposite direction.

Fake Traffic Injection

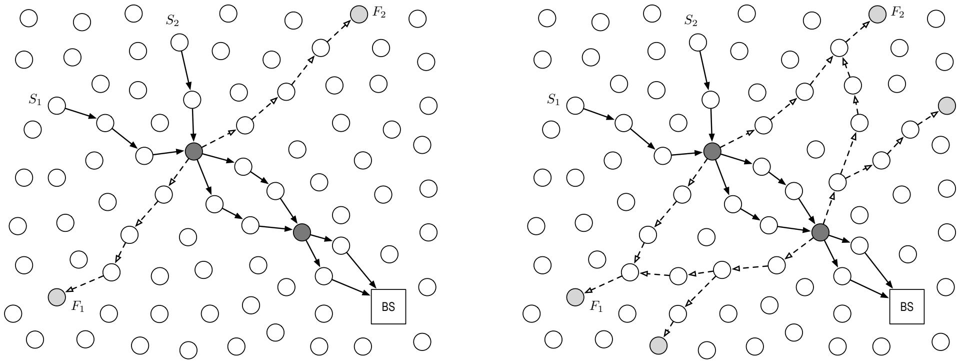
The aforementioned MPR solutions are still vulnerable to traffic analysis attacks since p_r is typically set to values over 0.5 for reasons of efficiency. Therefore, after a sufficient number of observations, an attacker can learn which of the neighbours of a node are its parents. To mitigate this problem, Deng et al. [38, 40] propose an additional technique called Fractal Propagation (FP) to be used in conjunction with MPR and RW. The main idea behind this mechanism is to generate and propagate fake packets in random directions in order to introduce

[12] A value $M_f = 1$ implies that the node receiving the fake packet does not retransmit the packet, which can be detected by the attacker.

more randomness into the communication pattern. When a sensor node observes that a neighbouring node is forwarding a data packet to the base station, it generates a fake packet with probability p_c and forwards it to one of its neighbours. The durability of fake packets is controlled by means of a global time-to-live parameter K. Also, if a node observes a fake packet with parameter k $(0 < k < K)$ it propagates another fake packet with time-to-live parameter $k - 1$. Figure 3.16d shows the trace resulting from the transmission of a single packet using the three mechanisms together.

The main problem of the FP scheme is that since nodes in the vicinity of the base station observe a greater amount of traffic, they generate much more fake traffic than remote nodes. This implies that the traffic rate in the area surrounding the base station is significantly higher than in other areas, which is not only detrimental to the operation of the network as it increases the number of collisions but also helps the adversary to track down the base station. To address this problem, the authors propose a new solution called the Differential Fractal Propagation (DFP). In this scheme, sensor nodes adjust their probability of generating fake traffic p_c according to the number of packets they forward. Below a given threshold sensor nodes behave as in FP but if their forwarding rates are higher (i.e., they are close to the base station) they reduce the probability p_c by a specific factor. Besides reducing the energy waste and packet loss rate, this scheme provides better privacy protection to the base station because it balances the network traffic load more evenly.

Yao et al. [153] devised another fake packet injection scheme to protect sink location privacy. In this scheme, real packets are sent along the shortest path from the data source to the base station. When two paths of real messages intersect at some point, the node receiving these packets sends two fake packets to two fake data sinks after a timer expires or a packet counter reaches a certain threshold. In this way, real and fake data sinks receive a similar number of packets. Moreover, when a packet reaches subsequent intersection points, the intersection node sends N_f packets to some random destinations. This process is depicted in Figure 3.17, where dark grey nodes represent intersection nodes, light grey nodes are fake sinks or some random data destinations. Ordinary arrows symbolise real data packets while dashed arrows represent fake packets. In Figure 3.17a the first intersection node transmits fake traffic to both fake data sinks. Meanwhile, the second intersection node introduces fake traffic to other random destinations as



(a) Injection at first intersection node (b) Injection at second intersection node

Figure 3.17: Yao et al.'s Fake Packet Injection Scheme

well.

The main problem of Yao et al.'s approach is its privacy protection level. An attacker starting from a data source and tracing packets can trivially reach the first intermediate node. From that point, the attacker has to decide on his next move. Since fake traffic is sent after certain conditions have been satisfied, the attacker can distinguish real from fake traffic. Additionally, since real data packets are sent using the shortest path, the transmission of fake traffic may imply an abrupt change in the angle of transmission and thus reveal the flow of real messages. This problem is also present in the Bidirectional Tree Scheme (see Section 3.2.1), which was devised by Chen and Lou [29] as a solution for protecting source- and receiver-location privacy simultaneously.

Sink Simulation

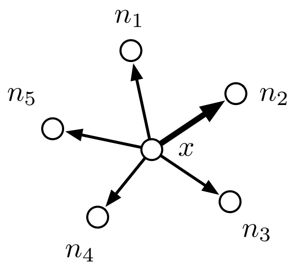
Some approaches try to emulate the presence of the base station at different points in the field in order to provide some form of k-anonymity[13]. Simulation techniques are based on the generation of fake traffic but, instead of being transmitted in random directions, it is addressed to particular network locations. This results in a concentration of high volumes of fake traffic, called hotspots, the objective of which is to draw the adversary to remote locations, away from the true data sink. The main challenge is to create hotspots that are evenly distributed throughout the network with a minimum overhead.

[13] Refers to the ability to remain anonymous (i.e., unidentified) within a set of at least k entities with similar attributes.

Chang et al. [25] present a solution called Maelstrom that generates a number of points in the network with a high traffic density which are intended to drag the attacker to them. After deployment, the base station sends N special configuration packets, each of which is configured to travel H_s hops away from the base station. After that, each of these packets travel H_r random hops to any node on the same level or further away. The final recipients of these packets become the centre of a maelstrom area and announce this by sending a discovery packet to nearby nodes. During data transmission, when a node receives a real packet it generates with probability p_f a fake message and forwards it to its closest maelstrom. Additionally, any packet addressed to the sink or a maelstrom is sent to a node closer to its destination with probability p and to a node at the same distance as itself (if any) with probability $1 - p$. By carefully adjusting the values of p_f, N, and p the authors claim that it is possible to evenly distribute the number of packets being received at the base station and the various maelstroms. However, once an intelligent attacker reaches a maelstrom area he can discard it as the true data sink.

A similar approach based on the simulation of several data sinks is proposed by Biswas et al. [17]. The idea is to evenly distribute multiple fake data sinks in such a way that each of them receive the fake traffic within its neighbourhood. The selection criteria is that fake data sinks should not be close to the base station, be neighbours with each other, or have neighbours in common. The goal is to maximise the number of neighbours that each of the fake base stations have, since this implies more incoming traffic. During data transmission, each node is configured to transmit a fixed number of messages either real or fake so that after a given time period all nodes have sent the same amount of traffic. Fake traffic is directed to fake base stations by its neighbours except for nodes which are not neighbours the selection of a fake destination is done in a round-robin fashion. The result should be that fake base stations receive at least the same amount of traffic as the actual base station. This approach may deal with naive rate monitoring adversaries but it can be easily defeated by informed global observers.

Finally, Deng et al. [38, 40] refined their fractal propagation solutions and created a new scheme called Differential Enforced Fractal Propagation (DEFP) that is capable of creating hotspots in a decentralised and dynamic way. To generate



neighs(x)	tickets	Probability
n_1	1	1/8
n_2	4	1/2
n_3	1	1/8
n_4	1	1/8
n_5	1	1/8

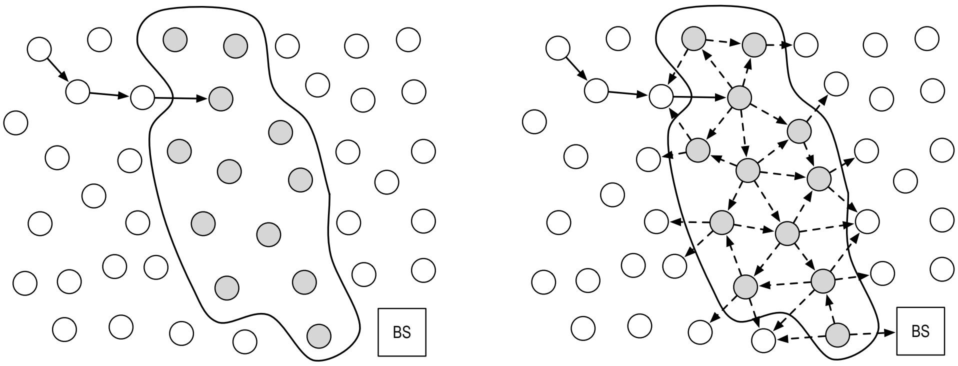
Figure 3.18: Decentralised Hotspot Generation in DEFP

hotspots, sensor nodes are pushed to send fake traffic to an already used neighbour with higher probability as opposed to FP and DFP where dummy packets are sent in any direction. This is achieved by keeping track of the number of fake packets forwarded to each neighbour. New fake traffic is more likely to be sent to neighbours who have previously received more fake traffic, as shown in Figure 3.18. In this way there is no need for a central authority or a complex coordination system to establish where the hotspots should be placed. Another interesting feature of this solution is that the hotspots can be deactivated by simply resetting the forwarding probabilities of each node. After that, new hotspot locations are likely to appear, which prevents smart attackers from discarding fake data sinks (i.e., hotspots) until they find the real base station.

3.3.2 Global Adversaries

The aforementioned techniques are considered to be effective only in a local adversarial model but some of them may also provide some means of protection against global adversaries. As a matter of fact, they can be useful if the global adversary has no real-time analysing capabilities, that is, he is only able to retrieve a snapshot of the amount of traffic transmitted over a period of time. Also, this is made possible by the fact that the adversary is usually unaware of the forwarding rate of each particular node rather he only knows the overall rate in its vicinity, as noted by Proano and Lazos [105]. However, there is still a chance that there can appear global adversaries with real-time monitoring capabilities, which needs to be tackled.

Again, the injection of fake traffic is one of the main approaches for protecting from global adversaries. Making the base station mimic the behaviour of sensor nodes, simulating the presence of several data sinks, and moving the base station to a different location might also be useful solutions. These schemes are usually



(a) Data packet reaches the backbone (b) Packet flooding within the backbone

Figure 3.19: Backbone Flooding

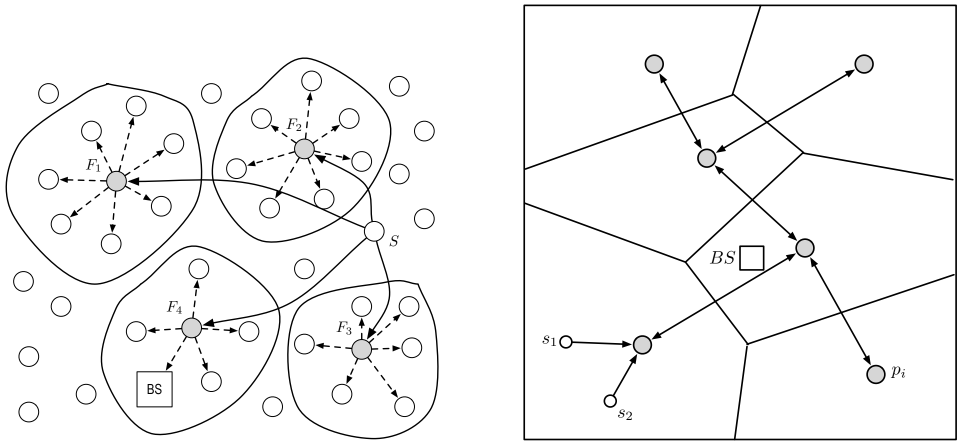
more energy efficient but imply more management and configuration issues as we see next.

Bogus Traffic

At the beginning of Section 3.3 we mentioned that flooding the network with messages is a simple yet efficient mechanism to homogenise network traffic and thus protect the location of the base station. The main drawback to flooding is the cost associated with the retransmission of the same message to every corner of the network. Backbone flooding [90] reduces the communication cost associated with flooding-based protocols by controlling the scope of the transmissions within a limited area, that is, among backbone members. Any data packet generated in the network is addressed to the backbone, from where it is delivered to all its members. Thereby, the backbone must satisfy two conditions. First, any data sinks must be located at least within the range of a backbone member in order to overhear all messages. Second, the backbone is created in such a way that it contains a sufficient number of nodes to achieve the desired level of privacy. A major limitation to this approach is that the backbone is static and thus backbone members will deplete their batteries sooner than the rest of the nodes. The authors suggest that this problem can be alleviated by (a) periodically rebuilding the backbone based on the energy remaining on the nodes or (b) defining several backbones from the beginning so that each packet is addressed to different backbones. Figure 3.19 illustrates the transmission of a data packet to the backbone as well as its eventual propagation and delivery to the base station.

Ying et al. also try to homogenise the traffic in the network by making each sensor node transmit at the same rate regardless of its distance to the base station. After network topology discovery, each sensor node knows its distance from the base station and can adjust its transmission rate accordingly. The Concealing Sink Location (CSL) [155] calculates the traffic that has to be transmitted by each single node located at distance i from the sink. This value is calculated as the number of nodes with distance $d \geq i$ divided by the number of nodes at distance i. This ratio represents the number of messages to be transmitted by each individual node at distance i, considering that each node must send its own traffic and forward the traffic from nodes further away from the sink. The number of nodes at a given distance i is estimated via geometric analysis considering the size of the deployment area and a uniform distribution of the nodes in the field. However, these estimations may differ significantly from the reality. Also, it is important to note that the authors assume that sensor nodes have a similar transmission rate for real messages but this might not be the case in the presence of bursts of messages.

A similar approach is followed in [156] to determine the transmission rate of sensor nodes, which is calculated based on the number of child nodes an immediate neighbour of the sink has. The reason is that this provides an estimation of the total amount of traffic that each node should generate to transmit a similar number of messages. The idea is to make all sensor nodes in the network transmit as many messages as a sink neighbour has to since they are the most loaded nodes. This fixed number of messages is split into real and fake messages. When a sensor node receives a message it first checks whether it is fake or real. In the former case, the packet is simply dropped, while in the latter, the packet is temporarily buffered before being transmitted. In the meantime the sensor node generates fake traffic to satisfy the overall transmission rate. Ying et al. claim that by instructing sensor nodes to forward the same number of messages as the neighbours of the base station, the lifetime of the network is not reduced. The argument is that the neighbours of the sink are always the first nodes to deplete their batteries. However, the authors have not considered several important issues that may call into question their claims. First, they should have considered that a transceiver in listening mode consumes almost as much battery as the micro-controller in a typical sensor node [132]. Second, sensor nodes must decrypt received packets in order to be able to discern which of them are real. Finally, it is necessary



(a) Mehta et al.'s sink simulation

(b) Network partition and tree formation

Figure 3.20: Examples of Sink Simulation Approaches

to consider that increasing the traffic rate of every single sensor node also has a negative impact on the reliability of the communications, which results in packet collisions and retransmissions.

Sink Simulation

Sink simulation has also been proposed as a mechanism to protect from global adversaries. Mehta et al. [90] propose simulating the presence of several data sinks in the field. During the deployment of the network some restrictions must be met. A number k of sensor nodes are picked as fake data sinks and the true data sinks are manually placed within the communication range of some of these. The number of fake sinks must outnumber the number of true sinks. When a source node collects event data, it sends them to all the fake data sinks, which on reception broadcast the message locally. This process is illustrated in Figure 3.20a, where the data source S sends four messages to $F_1, \ldots, F_2$ and each of them broadcast the message locally. Since all fake sinks receive the same amount of traffic, they are all equally likely to be next to a true data sink. The number of fake sinks has a clear impact on both the level of protection of the network and the communication overhead. The larger the value of k the better the protection but the higher the volume of traffic in the network.

Chai et al. [24] present a solution also based on the concept of k-anonymity. The idea is to have at least k nodes with a communication pattern similar to the nodes around the base station. To that end, the network is partitioned into k non-overlapping regions, each of which contains a node that collects all the information sensed in that region. These nodes p_i are organised as an Euclidean minimum-spanning tree (EMST) and the data they received from their own region is forwarded to all other tree members. The base station is manually placed after the formation of the EMST in order to ensure that it is within the communication range of the tree. The authors show in the paper that the Voronoi tessellation[14] of the network is the optimal partition that minimises the total routing energy and provides a reasonable protection level. Figure 3.20a shows a Voronoi partition of the network for the designated nodes p_i, in grey. Note that all nodes connecting the designated nodes see all the network traffic and thus the base station simply needs to be placed close to one of them. As a result, the uncertainty of the attacker is much greater than in the previous scheme for the same value of k. However, the nodes forming the tree are highly likely to deplete their batteries much sooner than the rest of the nodes, thereby ending up with no alternative routes to the base station.

Wang and Hsiang [141] propose another solution based on the creation of artificial hotspots that is intended to counter a global adversary. The hotspots are generated by means of a decentralised protocol that starts by generating a shortest-path tree rooted at the base station. After that, neighbouring leaf nodes from the tree can establish communication links in order to generate network cycles. During data transmission, the shortest-path tree is used to transmit data to the base station and, simultaneously, fake packets are injected into the cycles. Fake traffic continues moving along the cycle until it is completed. The centre of a hotspot is, indeed, a node where several cycles intersect as it is the recipient of all the bogus traffic generated along the cycles. Moreover, during cycle generation, they include a mechanism that establishes that two leaf nodes only create a cycle if their least common ancestor is at least h hops away from both nodes. This mechanism is interesting because it reduces the number of hotspots and, in this way, each of the hotspots receives a greater amount of traffic. However, if h is too large, it may result in very few hotspots, which turn out to be placed very close to the base station. Another drawback is that leaf nodes may be physically distant from each other and if this is the case they are unable to communicate

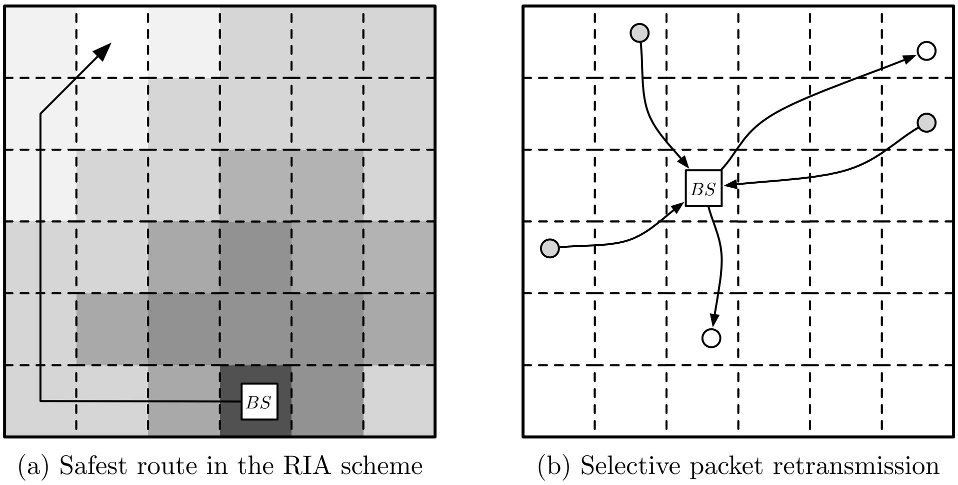
[14] A Voronoi diagram for a given set of locations is a partition of the plane into disjoint regions such that any given region contains all the points closest to each of the locations

with each other in order to establish a link for the cycle. Finally, it is worth mentioning that even though the authors assume a global adversarial model, this solution does not seem suitable for that purpose. As a matter of fact, their simulations concentrate on the communication cost and efficiency compared to DEFP but no security analysis nor simulation results are provided with respect to the level of privacy achieved by their solution. The main problem is that the true sink behaves differently from the rest of the artificial hotspots. While the transmission rate of the base station is negligible, fake hotspots must forward the real data packets coming from its child nodes. Consequently, the base station can be uncovered by calculating the node with the largest reception-transmission ratio.

Relocation and Disguise

As far back as 2003, Deng et al. [39] suggested the reallocation of the base station for enhanced security. They assume that the base station has complete knowledge of the topology of the network and thus it may calculate an optimal future location that maximises its security. Actually, they do not address a global eavesdropper but a compromised node dropping packets. Therefore, we refer the reader to their paper for further details.

Possibly motivated by the approach just mentioned, Acharya and Younis present the Relocation for Increased Anonymity (RIA) scheme [1]. The base station finds a new location by considering both the impact over network performance and its own level of protection. The network is divided into cells and the base station knows the transmission rate of each cell as well as the number of nodes in them. With this information, the base station calculates a score for each cell (i.e., $score_i = densitiy_i/threat_i$) and moves to the cell with the highest score. The rationale behind this scoring mechanism is that by moving the base station to a cell with a low threat (i.e., low transmission rate), the cells with high activity need to send packets to remote areas, which increases the delivery time and consumes more energy. Likewise, if there is a low transmission rate due to a reduced node density, moving the base station to that cell would cause the few nodes in the cell to become overwhelmed with traffic and their batteries would soon be depleted. Once the base station knows which is the most suitable cell to reside in, instead of moving there using the shortest path, the base station follows the safest route to reach the final destination. In Figure 3.21a we depict the path



(a) Safest route in the RIA scheme

(b) Selective packet retransmission

Figure 3.21: Relocation and Disguise Examples

selected by the base station for relocation based on the scores of each of its cells, the cells with higher scores are depicted in a lighter colour.

Mimicking the behaviour of ordinary sensor nodes is another way of hiding the base station from global adversaries. The communication pattern of the network can be modified by making the base station forward the packets it receives for several hops, as suggested by the Base-station Anonymity increase through selective packet Re-transmission (BAR) [1]. After receiving a packet the base station decides whether to send the packet to a random neighbour. Packets will be retransmitted away from the base station for a given number of hops. The length of the walk is dynamically adjusted based on the level of threat perceived by the base station. If the base station needs to increase its level of protection it defines longer walks. The general idea is that by doing this, the number of transmissions in remote cells increase and thus the attacker cannot clearly identify the actual location of the base station based on the transmission rate of a cell. An example of this approach is illustrated in Figure 3.21b, where source nodes and destination nodes are represented as grey and white circles, respectively. The main problem with this approach is that by forwarding packets to random remote locations, the base station is also increasing the transmission rate of the cells in its vicinity. Consequently, the attacker may still spot the base station as the cell with the highest transmission rate.

Finally, the Decoy Sink Protocol [35] combines indirection and data aggregation to reduce the amount of traffic received by the base station. Instead of sending the event data to the base station directly, sensor nodes are programmed to transmit their packets to an intermediate node (i.e., the decoy sink) and on their way the data are aggregated. Finally the decoy sink sends the result of the aggregation to the base station. Although this may prevent the attacker from determining the location of the true data sink, this scheme exposes the location of the decoy sink. If the goal of the attacker is to compromise the base station, he obtains a similar result by compromising the decoy sink. Also, if he destroys it, the protocol stops working. This problem is contemplated by the authors and they suggest picking several random nodes during the initialisation of the network to operate as decoy sinks. During the transmission period, sensor nodes send all their readings to a particular decoy sink for a pre-established period of time. This version of the protocol adds robustness to the network and balances the traffic load but the attacker is still able to ultimately achieve his original goal.

3.4 Conclusion

As a result of the aforementioned analysis we propose a complete taxonomy of location privacy solutions in WSNs (see Figure 3.22). This categorisation has been created following the same criteria as those used to guide the exposition of this chapter. It considers both the protection of node identity and traffic patterns as a first tier of the taxonomy. On the one hand, we observe that there are two main approaches for node identity protection, which are either based on the use of an already established pool of pseudonyms or rely on cryptographic schemes to generate them. On the other hand, the traffic pattern branch considers both the protection of data sources and the base station. In either case, we first classify solutions based on the capabilities of the adversary and then look into their common features. In general, we observe that the most common approach for protection against local adversaries is to introduce random routing paths. However, this type of protection mechanism is unable to preserve location privacy in the presence of an adversary with a global hearing range. Finally, little work has been done on the protection of location privacy against internal adversaries. In fact, there are no papers dealing with the threat of internal adversaries when the goal is to preserve the location of the base station.

Besides the information which can be easily derived from the proposed taxonomy, during the analysis of solutions we have discovered some other interesting

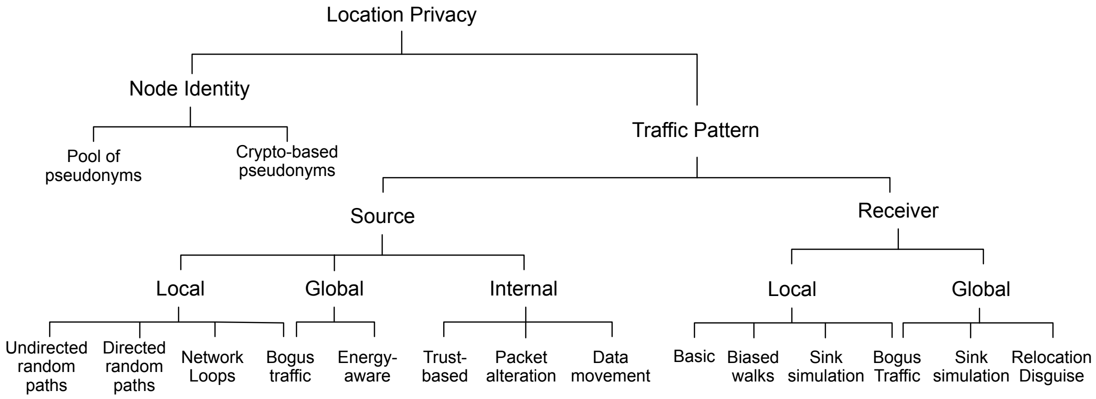


Figure 3.22: Taxonomy of Location Privacy Solutions in WSNs

issues that we will exploit in the following chapters to enhance location privacy in WSNs. First, we have observed that the existing protection mechanisms are blind, in the sense that they are executed without knowing whether an adversary is present in the field. With this sort of information, the network can intelligently decide when to activate the protection mechanism instead of assuming a constant threat. Moreover, if the network knows the exact or approximate location of the adversary it can carefully adjust the protection mechanism to impose a minimal impact on the network both in terms of energy consumption and data delivery delay. This is precisely the idea that we exploit in Chapter 4 to protect source-location privacy in the presence of a mobile adversary.

Another interesting observation is that most of the solutions devised to protect the location of the base station are either too costly because they require large amounts of fake traffic or they leak location information in some specific circumstances. Additionally, none of the existing solutions deal with node compromise attacks and more precisely with the threat of routing table inspection. An adversary being able to retrieve the routing tables of a node trivially learns which of its neighbours are closer to the base station. After very few repetitions of this process he gains a very good idea of the direction towards the base station. In Chapter 5 we elaborate on these research gaps and develop a solution that provides receiver-location privacy against adversaries capable of performing both types of attacks.

Chapter 4

Context-Aware Source-Location Privacy

This chapter presents a novel source-location privacy solution, called Context-Aware Location Privacy (CALP). The general trend towards source-location privacy protection has been to randomise routing paths in order to reduce the number of packets the adversary is capable of capturing, thus minimising his chances of tracing back to the source of messages. However, it is well known that sending packets on randomly chosen paths does not necessarily reduce the likelihood of the attacker reaching the source of events. The primary reason is that the data routing process is blind, that is, there is no knowledge of where the attacker could be located.

The CALP mechanism offers an original solution to the location privacy problem that takes advantage of the ability of sensor nodes to feel their environment. CALP exploits sensor nodes' context-awareness to detect the presence of a mobile adversary in their surroundings so that packets are routed in a more efficient and privacy-preserving manner. The solution aims to anticipate the movements of the attacker in order to minimise the number of packets he is able to capture and analyse, hence reducing the likelihood of the attacker finding the source. Unlike state-of-the-art solutions, the devised protection mechanism is operative only when the adversary is present in the field. Since the network is expected to be free from threats most of the time, the use of the CALP mechanism translates into significant energy savings and increased efficiency compared to previous source-location privacy solutions.

The remainder of this chapter is organised as follows. We describe the network and threat model under consideration in Section 4.1. The main building blocks of the CALP approach are detailed in Section 4.2. Section 4.3 presents the implementation of the shortest-path CALP routing algorithm, which combines the CALP approach with an energy-efficient routing algorithm. Finally, the shortest-path CALP routing is evaluated through simulations in Section 4.4.

4.1 Problem Statement

This section describes the network and attacker models considered in this chapter. It also presents the main assumptions that are relevant for the development in the CALP mechanism.

4.1.1 Network Model

We consider WSNs used for monitoring purposes that follow an event-driven data reporting method, meaning that individual sensor nodes transmit data packets to the base station as soon as they observe a relevant phenomenon in their vicinity. Therefore, all data is received at the base station after several forwarding hops.

The network is assumed to be composed of n sensor nodes which are uniformly and randomly distributed in a field. Sensor nodes cover a large area so that the attacker can only control and monitor a small portion of the communications at any given moment. Also, we assume that each node is aware of its adjacent neighbours and the connectivity of the network is high. This allows sensor nodes to choose the next communication hop from various neighbouring nodes.

The most important assumption for the correct operation of CALP is that each node in the network has the ability to detect the presence of moving objects in the field. This can be done by means of one or various types of sensors such as infrared, acoustic, thermal, pressure and magnetic sensors. Additionally, as shown in [157] and [146], the location of transceiver-free moving objects can be estimated due to the interferences they cause in the radio signal strength of several network nodes.

In addition, we require sensor nodes to share keys with its immediate neighbours in order to be able to encrypt and decrypt messages at each hop. We assume these cryptographic algorithms are semantically secure, thus enabling message confidentiality and indistinguishability to an external observer, who is unable to retrieve packet contents nor link messages. Moreover, the headers of the packets contain no information about the identity of the data sources. This can be achieved by means of pseudonyms schemes, as described in Section 3.1.

4.1.2 Threat Model

The adversarial model under consideration is an external, passive attacker with local eavesdropping capabilities. An external adversary does not control sensor nodes and thus cannot intercept packets and retrieve their contents. An adversary is said to be passive when he does not interfere with the communications or the normal operation of the network by injecting, modifying or blocking packets. In general, passive adversaries limit their actions to performing traffic analysis attacks. These attacks depend on the hearing range of the adversary, which is typically equivalent to that of an ordinary sensor node. This must not be regarded as a strong assumption since the network model under consideration is intended to cover large areas.

Also, contrarily to traditional attackers considered in [108, 109], the adversary is able to move in the direction of received packets. An attacker is able to determine the angle of arrival of a signal, for example by measuring the difference in received phase at each element of an antenna array [86], which finally allows him to find the source of a packet. Besides, we assume that the attacker is able to move at a reasonable speed but never exceeds the time it takes for a packet to reach a neighbouring node. Thus the speed of the attacker is not a critical factor, although it affects the response time of our scheme.

The attacker might start to monitor the communications from either an internal position or the edge of the network. We follow the same approach as most authors, that is, letting the adversary start next to the base station. In this way, the adversary will eventually overhear data packets since all the traffic is addressed to this single node. We consider two different strategies for the adversary: a patient or inquisitive adversary. Formally,

Definition 1 ($\mathcal{ADV}_{\mathcal{PAT}}$)**.** *Let* $X = \{x_1, x_2, \cdots, x_n\}$ *be the set of sensor nodes comprising the network and let* x_0 *be the base station.* $\mathcal{ADV}_{\mathcal{PAT}}$ *is an attacker that starts at* x_0 *and waits until he observes a packet from another node* x_i*. The adversary moves to node* x_i *and waits for a new transmission from a node* x_j*,* *where $i \neq j$. If no packets are received after a time t, the attacker returns to node x_i. This process is repeated until $x_i = x_0$.*

The patient adversary waits until he overhears a data packet or a predefined time period passes without any observations, in which case he returns to his previous position. Eventually, the attacker may return to the original position, the base station. The inquisitive adversary behaves similarly but he does not wait for packets.

Definition 2 ($\mathcal{ADV}_{\mathcal{INQ}}$)**.** *Let $X = \{x_1, x_2, \cdots, x_n\}$ be the set of sensor nodes comprising the network and let x_0 be the base station. $\mathcal{ADV}_{\mathcal{INQ}}$ is an attacker that starts at x_0 and initiates a random walk until he observes a packet coming from node x_i. The adversary waits next to node x_i for new packets but if no packets are received after a time t, he initiates a new random walk.*

A combination of both strategies is also possible but the proposed solution will be only evaluated against the $\mathcal{ADV}_{\mathcal{PAT}}$ and $\mathcal{ADV}_{\mathcal{INQ}}$ adversaries.

4.2 Context-Aware Location Privacy

This section provides the details of the Context-Aware Location Privacy scheme. The basic idea behind the CALP mechanism is to anticipate the movements of the attacker in order to decrease the number of packets he is able to capture and thus reduce the probability of the attacker finding the data source. To that end, it is necessary to take advantage of the ability of sensor nodes to perceive the presence of moving objects in their vicinity. Upon the detection of such an event, nodes react by broadcasting a route update message to its neighbouring nodes. This message is forwarded several hops away from the position of the attacker and is used to modify the routing tables of the nodes in such a way that packets are routed around the region under the control of the adversary.

4.2.1 Software Components

The Context-Aware Location Privacy scheme can be regarded as a software plug-in that integrates neatly with the rest of components of the sensor nodes to enable privacy-aware routing protocols. The interaction between components is depicted in Figure 4.1, where an outgoing arrow means that the component uses some of

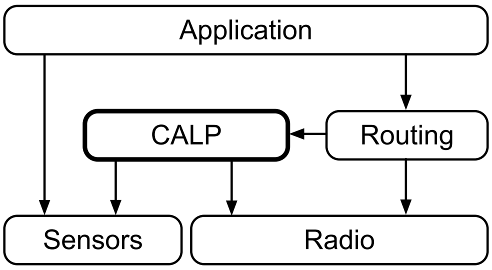


Figure 4.1: Components Interdependence

the functionality provided by the component receiving the arrow. Therefore, a monitoring Application might use the Sensors component to measure some phenomena and a Routing component to send the information to the base station. Additionally, the Routing component uses the Radio component to send the data through the wireless interface and might use the CALP component to make decisions on the next hop of the communication, thus allowing the sensor nodes to adapt their routing strategy depending on their privacy needs. Finally, the CALP component may use either the Sensors component or the Radio Component, or both, to detect the presence of adversaries.

The main advantage of this approach is that by integrating the CALP component, any existing application can transparently benefit from privacy-enhanced routing. Moreover, the underlying routing protocol does not need to be modified or replaced by a specially tailored solution since the interaction between the Routing and the CALP component is done seamlessly through an intermediate shared element, i.e., the routing table of the node. More details will be given in the subsequent sections.

4.2.2 Adversary Recognition

Prior to the route updating process, the network must identify whether there is an adversary in the field. This implies that the CALP mechanism is suitable for application scenarios where the tracking of moving objects is among the typical duties performed by the sensor nodes. The monitoring of endangered species, the surveillance of country borders, mineral deposits or oil and gas fields are among the scenarios where sensor nodes already incorporate the object tracking functionality. Most of these scenarios are highly sensitive to the presence of intruders and the authorised-personnel-only policy must be enforced.

The use of traditional radio-based localisation methods [86], where the target object carries a transmitter or transceiver whose radio signals are analysed to determine its location, are not suitable for critical object tracking scenarios because an intruder might not have such a device or can simply drop it. Also, the use of physical barriers have been a means of protection but in some cases, such as a country's perimeter surveillance, this might be highly expensive or even infeasible. Given such circumstances, the use of WSNs capable of detecting and tracking objects crossing the area under observation is of great interest [69, 77]. To that end, the nodes comprising the network can be equipped with motion sensors or they might measure the interferences in the signal strength of the radio signals [146] caused by the moving objects.

The aforementioned techniques allow sensor nodes to determine the existence of mobile targets in their vicinity. However, these techniques on their own provide no means of discriminating between adversaries and authorised users or other moving objects. As a matter of fact, being able to distinguish adversaries from other mobile entities is not a trivial task. The only difference is between entities authorised to move around the field (e.g. those being monitored or network administrators) and other moving objects, which may be adversaries or not. Therefore, the best strategy for the sensor network is to consider that any non-authorised moving object is an adversary although, ideally, the protection mechanism should be launched only in the presence of adversaries in order to reduce the extra overhead due to the performance of the privacy-aware routing mechanism. Anyway, this strategy is much more energy-preserving than already existing solutions, which are in continuous operation.

Consider a sensor network which monitors the behaviour of an endangered animal species. This network needs to be able to distinguish between different species so that it collects only relevant information concerning the protected species. This can be done in several different ways, for example, by tagging the animals with some sort of wireless device (e.g. an under-skin transmitter) being able to broadcast authenticated information regarding each specific animal. Also, biologists might carry their own personal devices in order to be recognised as authorised users. On the other hand, other animal species or adversaries willing to capture the protected animals would trigger the protection mechanism as they are not in possession of a legitimate device.

A simple challenge-response protocol might allow the interaction of external authorised entities with the sensor network. After authentication, a temporal session key might be established between the sensor network and the external entity in such a way that this entity is able to securely transmit messages to the sensor network. Clearly, the session key must be occasionally updated. This process may require the use of public key operations. Several solutions have been devised for the user-authentication problem [28, 136]. Also, similar solutions exist for unattended WSNs, where the sink sporadically visits the field to collect data from every single node [41]. Doubtlessly, the advances in Elliptic Curve Cryptography will not only simplify the process but also reduce the overhead introduced by the use of authentication mechanisms.

Also, in order to reduce the probability of erroneously identifying moving objects as adversaries, the sensor network might observe the behaviour of moving objects in the field in the presence of messages. Therefore, if a non-authorised moving object is detected by the network, the sensor nodes in the vicinity of the mobile object might mimic source nodes and send out fake messages. In the case the moving object traces back fake messages it is highly likely to be an adversary and thus the sensor nodes might alert their neighbours to this, by broadcasting route update messages.

4.2.3 Route Updating Process

Upon the detection of an adversary in the proximities of the network, the devised privacy-preserving mechanism is triggered. The sensor nodes feeling the presence of an adversary, inform their neighbours of the situation in order to prevent packets from traversing the area where the adversary is located. As the adversary is capable of moving in any direction, it is also necessary to anticipate his movements in order to minimise the number of packets he might be able to capture. Thus, alert messages need to expand over several hops so that it is not only neighbours in a close range to the attacker that are aware of the distance to the adversary.

After the detection of an adversary in the field, the detecting node (at distance 0) informs their immediate neighbours about the presence of the adversary by broadcasting an alert message with distance value 1. Upon reception, the receiving nodes store the distance value and broadcast a new message with distance value 2. This process is repeated for a given number of hops to spread the alert. Clearly, the number of hops the alert spans depends on the ability of the attacker

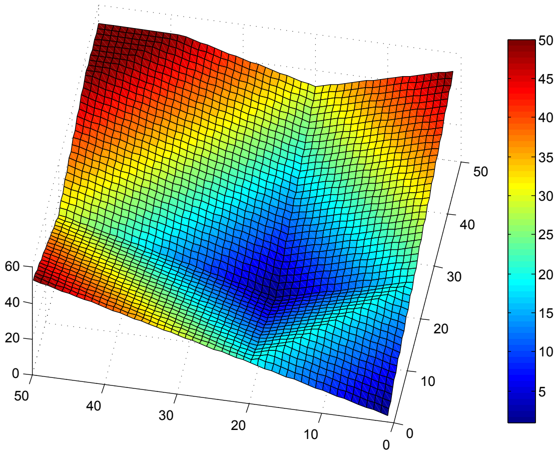


Figure 4.2: Distance of sensor nodes with respect to two adversaries

to monitor the communications. The more powerful the attacker, the larger the radius of the area covered by the routing update message. Figure 4.2 depicts the distance values obtained by each sensor node in a network of size 50 × 50, where two adversaries have been detected at positions $(20, 20)$ and $(0, 0)$.

The power of the attacker can be measured by two non-mutually exclusive means, namely, the communications area the attacker is able to monitor and the displacement speed. In the work presented here, we focus on mote-class attackers, which are capable of eavesdropping and analysing the traffic in a region r equivalent to that of any regular sensor node. This feature is also dependent on the size of the network since a large network is less vulnerable to an attacker with a hearing range of r than a network covering a small region. With regards to the speed of the attacker, it is important to note that an adversary moving at an infinite speed has the ability to capture every packet in the network. Obviously, this type of attacker is unrealistic and thus is beyond the scope of our solution. We assume that the network is agile enough to reconfigure the routing tables before the attacker reaches the next neighbouring node. In fact, this is not such a strong assumption since the time of flight of packets between contiguous nodes and the processing time of packets can be considered negligible.

Whenever a detector node sends a route update message to its neighbours, this message contains information regarding the distance at which the attacker is placed. In general, the number of hops is a good indicator of the distance if the sensor network is uniformly deployed, though sophisticated devices might provide more precise information about the location of the attacker. Nevertheless, using a hop-based distance estimation simplifies the route updating process because upon the reception of an update message, the receiving node merely increments the hop count before forwarding the packet and the routing table is modified in consequence without having to perform any further calculations to determine the distance between the node and the adversary.

Finally, it is worth noting that route update messages do not provide the adversary with information regarding the location of the data sources because their transmission is independent of the presence of events in the network. Therefore, these messages may be either sent periodically or just in the presence of adversaries but the latter choice is recommended to extend the lifetime of the sensor nodes. An alternative is to benefit from beacon frames, which are configuration messages that are periodically broadcast regardless of the existence of events. Beacon frames have the ability to carry a few bytes of information in the payload, which is enough for alerting about the distance to the adversary. Beacon frames do not imply any extra energy consumption in the network and thus allow resources to be saved. However, this approach has some limitations in terms of the delay between two consecutive frames, which ranges from tens of milliseconds to hundreds of seconds, as described in [128]. Therefore, there is a trade-off between the energy consumption and the routing update speed, which impacts on the privacy preservation of the source nodes in the case of having to counter rapidly moving adversaries.

4.2.4 Data forwarding

The data forwarding process is dependent on the underlying routing algorithm used to transmit event data from the source nodes to the base station. The CALP mechanism can be regarded as a plug-in component that modifies the routing tables of sensor nodes in such a way that the selection of the forwarding nodes is conditioned not only by their distance from the base station but also by their distance from the adversary. Thus, upon receiving a data packet directed to the base station, the recipient node decides in which direction to forward the message based on the routing strategy and, additionally, the distance from its neighbours to the attacker. These data are stored in the routing table of each node.

There are at least two options when sending packets to nodes which are located at a close distance to the adversary. One might choose to impede sensor nodes from forwarding packets to those neighbours located at a distance of less than a *minimum safety distance* from the adversary, that is, data packets must circumvent the region where the adversary is. On the other hand, instead of simply blocking the arrival of data packets to sensor nodes in the proximities of the adversary, we might choose to penalise the selection of these nodes with respect to other neighbours outside the established minimum safety distance. We refer to these two strategies as strict and permissive data forwarding.

The use of a *strict* safety distance has the advantage of ensuring that the attacker will not capture any packets unless he moves fast enough to cover areas at a distance greater than the predefined minimum safety distance. We assume that adversaries are incapable of moving that fast (see Section 4.1.2). Nonetheless, the use of a strict security perimeter presents some drawbacks that might negatively affect the operation of the network. Specifically, the greater the minimum safety distance, the greater the number of hops a packet will traverse in the presence of an adversary in the proximities of the communication path. Consequently, the delivery delay and the overall energy consumption of the network will increase. This might also result in the non-delivery of data packets at the base station if the adversary is in its vicinity and the security perimeter is sufficiently large. In that case, data packets travel back and forth originating network loops until the adversary moves to another region. A possible countermeasure to this problem is to make sensor nodes temporarily store any received data packet, but if the adversary being countered is patient, i.e., he does not move until the reception of a data packet, the delivery time significantly increases. Also, if the sensor nodes continue to receive data packets they might run out of memory and therefore, they should turn to dropping some packets.

On the other hand, a *permissive* security perimeter avoids the need of buffering data packets at intermediate sensor nodes in the vicinity of an adversary, thus saving memory and reducing the delays in the delivery process. Thus, a permissive minimum safety distance is more suitable for real-time applications while the strict version is convenient in delay-tolerant application. Notwithstanding, a permissive security perimeter provides a lower privacy protection level since data packets may be forwarded to nodes placed within the hearing range of the adversary. As a result, the adversary is more likely to reach the data source. Clearly, there is a trade-off between overhead and privacy protection associated with the data forwarding strategy. Further analysis and discussion is provided in

Section 4.4.3.

4.3 Shortest-path CALP Routing

The CALP mechanism can be used in conjunction with different routing protocols to enhance source-location privacy protection. Although this mechanism can be applied to any routing protocol, here we focus on the application of CALP to a shortest-path routing algorithm since they provide interesting features such as minimal latency and reduced energy consumption.

4.3.1 Shortest-path routing

Several shortest- or single-path routing techniques can be found in the literature [62, 95]. These energy-efficient routing protocols allow sensor nodes to deliver data packets to the base station using the minimum number of neighbours as data relays. Whenever a node has data to transmit, it picks the neighbour node which is closest to the base station and sends the packet to it. The recipient repeats the process until the packet is eventually received at the data sink. Since each sensor node always picks the neighbour that is closest to the destination, the path followed is the most energy-efficient and it also incurs the shortest delay.

Usually, these techniques require that either sensor nodes are equipped with additional hardware or that an initialisation phase is performed. The simplest way to enable this sort of routing protocol is by means of a topology discovery protocol, where the base station floods the network with a distance value initially set to 0 that is incremented at each hop; similar to the route updating process described in Section 4.2.3.

The shortest-path routing technique considered in this section makes greedy forwarding decisions since it selects locally optimal neighbours. A neighbour is considered to be locally optimal when it minimally deviates from the straight line connecting the data sender and the destination. An example is given in Figure 4.3, where N represents the node sending data in the direction to the data sink (S) and A, B, C, D, E are the neighbours of N ($neighs(N)$). Also, $\alpha = \angle NAS$, $\beta = \angle NBS$, and $\gamma = \angle NCS$ are the angles formed between the line $\overline{NS}$, and $\overline{NA}$, $\overline{NB}$ and $\overline{NC}$, respectively. For the sake of simplicity only some of the angles have been represented. Thus, X is the locally optimal neighbour of N if $\forall X, Y \in neighs(N) \wedge X \neq Y, \angle NXS \leq \angle NYS$.

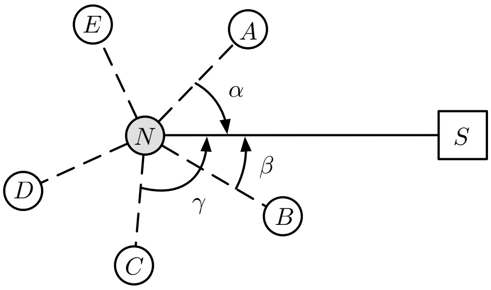


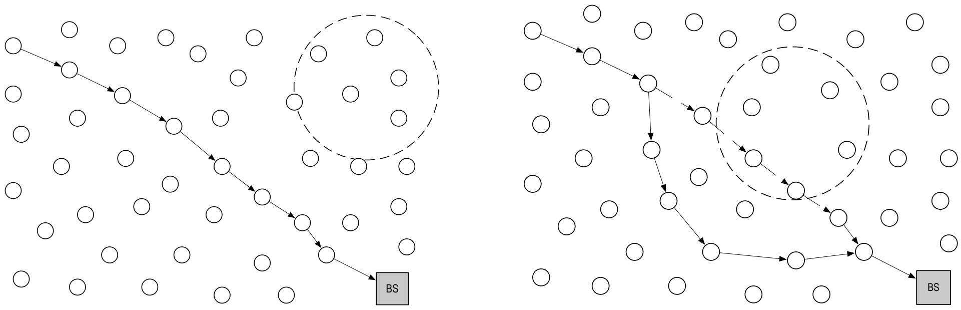
Figure 4.3: Locally optimal neighbour selection

The main advantage of implementing a greedy shortest-path technique is that only a small amount of internal storage is required in the nodes to operate. In order to route data packets, a sensor node needs information about its own neighbours and the location of the base station, but it does not have to be in possession of information about other intermediate nodes. The main limitation of a greedy approach is that the path followed by the packets might not be globally optimal even though it is locally optimal, i.e. there might exist more efficient paths.

4.3.2 Combination with CALP

When combined with the CALP mechanism, the greedy shortest-path routing technique acquires the ability to anticipate the movements of the adversary in such a way that the number of packets he might be able to capture is significantly reduced. Additionally, the packets will minimally deviate from the shortest path to the destination, thus the extra energy consumption incurred by the operation of our privacy preservation mechanism is notably reduced compared to other solutions. Moreover, note that the deviation from the most energy-efficient path only takes place when the adversary is located close to that area. Figure 4.4 depicts a scenario where the network adapts the routing path in order to circumvent an adversary moving in the vicinity of the shortest path. The area controlled by the adversary is represented as a dashed circle while dashed arrows represent a temporary suppression of messages.

Two versions of the shortest-path CALP routing have been devised. In the first version, a strict minimum safety distance is considered. Consequently, the route update messages are used to create an impassable security perimeter, which data packets never traverse. When the distance from the adversary to the shortest



(a)

(b)

Figure 4.4: Path adaptation depending on the presence of the adversary

Algorithm 1 Sending strategy: Strict CALP routing

```
Input: $MIN\_SAFETY\_DIST$
Input: $data$
 1: $neighs \leftarrow get\_neighbours()$
 2: for all $n_i \in neighs$ do
 3:     if $distance(n_i) \leq MIN\_SAFETY\_DIST$ then
 4:         $penalty[n_i] = \infty$
 5:     else
 6:         $penalty[n_i] = angle(n_i) + \pi/distance(n_i)$
 7:     end if
 8: end for
 9: $next\_hop \leftarrow minimum(penalty, neighs)$
10: $send(data, next\_hop)$
```

path is shorter than the minimum safety distance, i.e., the adversary is over the shortest path, data packets will deviate from the original path to avoid crossing the security perimeter. This behaviour is described in Algorithm 1. Basically, whenever a sensor node has data to transmit or forward, the node obtains a list of neighbours from its routing table and for each of them calculates a penalty based on their distance to the adversary. This penalty is maximum when the neighbour is in range of the adversary (lines 3-5) but it is a linear function of the distance and the deviation from the shortest path otherwise (lines 5-7). Finally, the data packet is sent to the neighbour with the lowest penalty.

In the permissive version of the protocol, data packets do not necessarily change their route in the case of an adversary placed in the shortest path. Packets are only deviated if the cost associated with performing such a choice is greater than the cost of entering the adversary's hearing range. A detailed description of this behaviour is provided in Algorithm 2. Similar to the strict version, the

Algorithm 2 Sending strategy: Permissive CALP routing

```
Input: $MIN\_SAFETY\_DIST$
Input: $data$
1: $neighs \leftarrow get\_neighbours()$
2: for all $n_i \in neighs$ do
3:     $penalty[n_i] = angle(n_i) + \pi/distance(n_i)$
4:     if $distance(n_i) \leq MIN\_SAFETY\_DIST$ then
5:         $penalty[n_i] = penalty[n_i] + 1/distance(n_i)$
6:     end if
7: end for
8: $next\_hop \leftarrow minimum(penalty, neighs)$
9: $send(data, next\_hop)$
```

algorithm is activated when a node has data to transmit or forward. The node obtains the list of neighbours and for each of them it calculates a base penalty (line 3), which is incremented by a factor that is inversely proportional to the distance of the neighbour to the adversary (lines 4-6) in the case the adversary is in its vicinity. The neighbour with the lowest penalty is finally chosen to receive the data.

As previously described, when the adversary is not present in the field, the proposed algorithms must behave as the original shortest-path routing protocol. Therefore, the locally optimal forwarding neighbour is chosen so that it minimally deviates from the straight line connecting the data sender and the base station. To that end, the distance to the adversary is used as a penalty value in such a way that the closer the adversary, the greater the penalty. In particular, we penalise the proximity of a neighbour to the adversary exactly $\pi/distance$ units. Depending on whether the version in use is strict or permissive, an additional penalty is introduced when the distance to the adversary is less or equal than the predefined minimum safety distance. The minimum safety distance is a parameter of the solution (MIN_SAFETY_DIST) that might be tuned by the administrator of the network to carefully balance between privacy protection and usability.

Finally, note that both algorithms are based on straightforward operations that can be performed even by extremely hardware-constrained devices. Additionally, the CALP requires some extra memory in order to store information about the distance from the adversary to each of the neighbours. Table 5.1 shows the routing table of a particular node, where the right-most column has been

neighs	*angle*	*distance*
A	$\pi/4$	2
B	$\pi/5$	4
C	$5\pi/9$	5
D	$8\pi/9$	3
E	$11\pi/18$	4

Table 4.1: Routing table of node N

added to keep distance information. These values are updated upon the reception of the route update messages described in Section 4.2.3.

4.4 Protocol Evaluation

In this section we evaluate the performance and privacy protection level of the proposed shortest-path CALP routing mechanism. First, we briefly describe the simulation scenario.

4.4.1 Simulation Scenario

We developed a discrete-event simulation environment in MATLAB and conducted extensive experiments on it. The simulator enables multiple simultaneous transmissions from various data sources as well as the presence of various local adversaries moving in the field. The simulator obviates the low-level communication problems (e.g., collisions) and focuses on the application and routing layers since our goal is to demonstrate the feasibility of the proposed solution.

The setup used for our simulations is similar to that commonly found in the literature [60, 128]. We deployed a large wireless sensor network consisting of $n \times n$ uniformly distributed nodes, where $n = 100$. Each simulation instance is run 50 times and each of the instances consists of 500 simulation steps. A new data message is generated and forwarded by the data source at each simulation step. Also, a beaconing phase is scheduled so that the network is aware of the whereabouts of the adversary and thus packets are routed accordingly. Source nodes are placed at different distances from the base station but are static during each simulation.

In the first simulation step the adversary is placed in the proximities of the base station, which is located at the centre of the network by default. The adversary under consideration is either inquisitive or patient. The inquisitive adversary moves randomly until he overhears a transmission in his vicinity, in which case he moves in the direction of the received message. If he follows a trace of packets and after a period of time no message arrives at his current position he starts to move randomly in the search of new packets. On the other hand, the patient adversary only moves in the presence of packets in his vicinity. Moreover, adversaries might move at different paces with respect to the simulation steps, however the speed is fixed and constant within a single simulation instance.

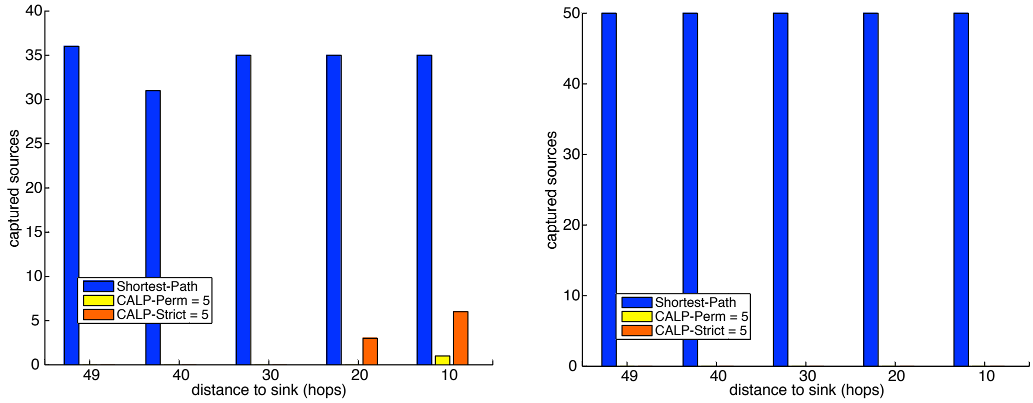
By default, the network safety distance is set to 5 meaning that communication paths are modified when the adversary is closer or equal to that distance from the shortest path. Furthermore, the hearing range of the adversary has been set to a monitoring radius equivalent to that of a sensor node. Although the simulation environment allows for several simultaneous adversaries in the field, we study the effect of a single adversary. Note that the simulations were conducted such that the adversary is considered to be in the field at all times. However, in real scenarios this is not the case, the adversary enters and leaves the network at will.

The simulation ends under two circumstances, either when the adversary reaches the source or when the last simulation step is reached without the adversary being able to find the data source.

4.4.2 Privacy Protection

This section evaluates the privacy protection level provided by both the permissive and strict versions of the shortest-path CALP routing scheme. The level of protection is measured as the number of source nodes the adversary is able to capture in each simulation instance. The two versions of the proposed mechanism are compared with each other and with respect to the traditional shortest-path routing scheme for various source-sink distances. Moreover, the simulations are conducted in the presence of both inquisitive and patient adversaries. The results are depicted in Figure 4.5 as a bars diagram where the x-axis represent the distance to the base station and the y-axis show the number of total number of captures after 50 simulations of each instance.

From the simulations we observe that the distance of the source node with respect to the base station has no clear impact on the privacy protection level. The adversary is able to reach the data source in roughly the same number of



(a) Inquisitive Adversary

(b) Patient Adversary

Figure 4.5: Number of captured sources

cases regardless of the distance. The patient adversary (see Figure 4.5b) always reaches the source node because he waits next to the base station until he receives a packet addressed to it. After that, since all packets follow the same path, he reaches the data source in the minimum number of steps. On the other hand, the inquisitive adversary (Figure 4.5a) is less likely to find the data source even for a single-path routing algorithm. The main drawback for inquisitive adversaries is that at the beginning of the simulation they might move away from the original location thus missing some of the packets arriving at base the station several simulation steps later. These adversaries are only successful if at some point during the simulation they come across with the communication path.

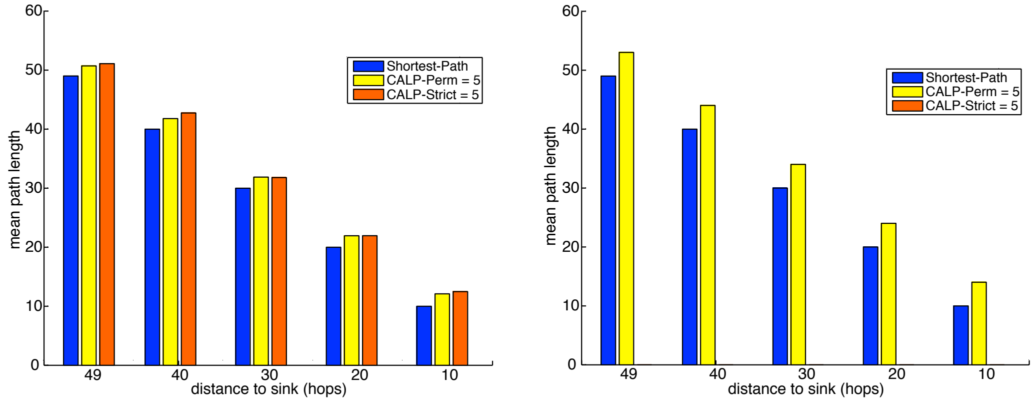
When the shortest-path routing protocol is used in conjunction with the CALP mechanism, the situation improves enormously. When the adversary is inquisitive (see Figure 4.5a), he is only capable of compromising source-location privacy when the distance between the data source and the base station is relatively short. Surprisingly, the permissive version of the shortest-path CALP routing provides better protection level than the strict version. The reason is that, in the strict version, the movements of the adversary are never conditioned by the packets traversing the network since he is not able to overhear them given a sufficiently large safety distance, as the one being used. In the permissive version, an inquisitive adversary is able to overhear some of the packets because under certain circumstances the nodes might choose a node within the security perimeter as the next hop. This causes the adversary to move in the direction of the received packet but since the path changes dynamically based on his movements, he might overhear packets coming from different neighbouring nodes which misleads him from the target.

When countering a patient adversary (see Figure 4.5b), neither version of our protocol ever leaks location information about the source node. Apparently the packets are able to circumvent the attacker without being detected. In the permissive version, the packets might reach the base station by traversing the safety distance thereby causing the patient adversary to move towards those packets. Being the adversary in a new location away from the base station, the new paths are re-adapted thus being able to circumvent the adversary and reach the base station. However, in the strict version, since the adversary is initially placed next to the base station and the packets are not allowed to traverse the safety region, the task of delivering the packets to the base station is not fulfilled. This issue is reviewed in more detailed in the following sections.

4.4.3 Protocol Performance

We evaluate the performance of the protocol by means of the length of the resulting routing paths. The length of the path not only determines the delivery time of the packets but also the overall energy consumption of the network. Larger paths result in more transmissions and consequently have a negative impact on the lifetime of the batteries. In general, single-path routing algorithms are considered energy-efficient algorithms since data packets are sent via the shortest path from the source node to the base station. However, these algorithms provide the lowest protection level as all the packets follow the same (shortest) path. Therefore, the inclusion of the CALP mechanisms to a shortest-path routing scheme effectively trades off between performance and privacy protection level.

The mean path length of the packets travelling from the source nodes to the base station is represented in Figure 4.6. In general, the mean path length is slightly higher than the minimum expected value, i.e. the length of the path originated by the shortest-path routing algorithm. Clearly, in the presence of an inquisitive adversary (see Figure 4.6a), the permissive version of our scheme provides better results than the strict version. On the other hand, for a patient adversary (see Figure 4.6b), the permissive approach originates paths that are on average slightly longer than those generated in the presence of an inquisitive adversary. The reason is that for a patient adversary the nodes need to deal with an adversary waiting in the vicinity of the base station. More importantly,



(a) Inquisitive Adversary

(b) Patient Adversary

Figure 4.6: Mean path length

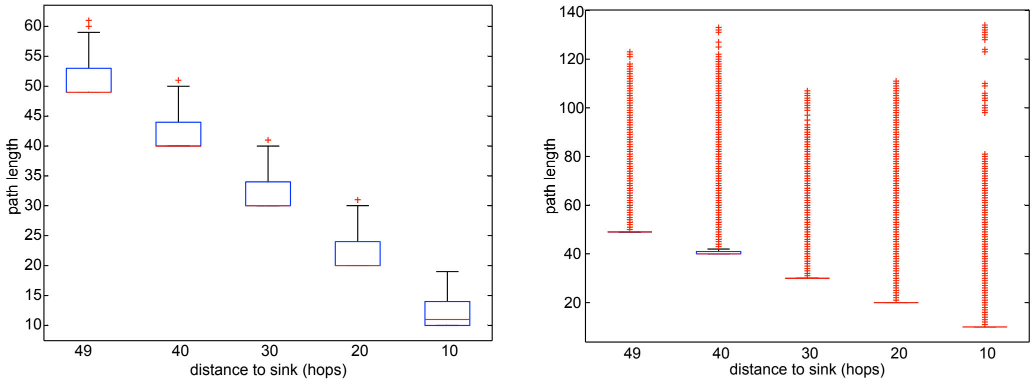
the strict version is unable to generate data paths that circumvent the patient adversary and eventually deliver the packets to the data sink. Since the length of the packets is stored when they arrive at the base station, no data is shown for this scheme in Figure 4.6b. This problem is not due to the design of the CALP mechanism but is caused by the particular strategy followed by this type of adversary.

This problem can be lessened in several ways depending on the requirements of the network. In a sensor network with no real-time requirements (i.e. it tolerates moderate latencies), instead of sending messages back and forth at the border of the security perimeter, intermediate nodes could temporarily store the packets until the adversary decides to move away from the base station. However, if the adversary is patient enough, the highly constrained memory of the sensor nodes would require some of the packets to be dropped. Therefore, a more convenient approach to deal with this issue is to implement a mixed version of the CALP mechanism including the benefits of both the permissive and strict schemes. The idea is to switch from a strict to a permissive approach as packets approach to the base station. In this way, if a patient adversary is next to the base station using a permissive strategy attracts the adversary away from it and allows the delivery of packets. In addition, using a strict strategy is expedient when the adversary is close to the data source because at that point capturing a few packets might lead to the target. Also, it is possible to overcome the problem by dynamically re-adapting the safety distance depending on the whereabouts of the adversary or by switching from the CALP approach to one of the solutions based on the creation of random routes, such as the Phantom Routing.

Despite the mean path length being close to the minimum value, some isolated packets might traverse a large number of intermediate nodes before being delivered. Therefore, studying mean values is not enough and next we look into the path length distribution. In Figure 4.7 we present the path length distribution in the presence of an inquisitive adversary with box plots[1], which is a useful way of describing the degree of dispersion and skewness in the data, and identifying outliers. On the left-hand side of the figure, we can observe that when using a permissive security perimeter, most of the packets travel a similar number of hops before reaching the destination. The mean value is very close or equal to the distance to the sink and there are only a few packets that travel long distances (outliers). However, the landscape changes dramatically when using the strict version of the scheme, as shown on the right-hand side figure of the figure. Some isolated packets may travel up to 134 hops before reaching the base station. The reason for such long paths is the creation of network loops due to the presence of the adversary in regions close to the sink. Packets are sent in the direction of the base station but nodes on the border of the security perimeter cannot send them forward and choose to relay them to other nodes which are in the same situation. Finally the packets are returned to any of the nodes which initially sent those packets. Keeping a list of already seen packets could help avoid network loops, however, since the adversary is able to move, the next time a node receives the packet the situation might be different, i.e. the direction, which was previously occupied by the adversary, might currently be safe.

In general, we can claim that the permissive version provides an adequate protection level without incurring an excessive overhead to the network. It is true that in a few special cases the protocol generates paths that are a slightly longer than usual but this is not too problematic. However, we acknowledge that the overhead incurred by the strict version of the protocol might be overly high. Consequently, this strategy must be used only when the criticality of the scenario demands an extraordinary privacy protection level. Notwithstanding, a mixed strategy might be the best option to keep a reasonable path length.

[1] In each box, the central mark is the median, the edges of the box are the first and third quartiles, and the whiskers extend to the minimum and maximum data points or to 1.5 times the interquartile range. Outliers ('+') are values behind the whiskers.



(a) Permissive CALP (b) Strict CALP

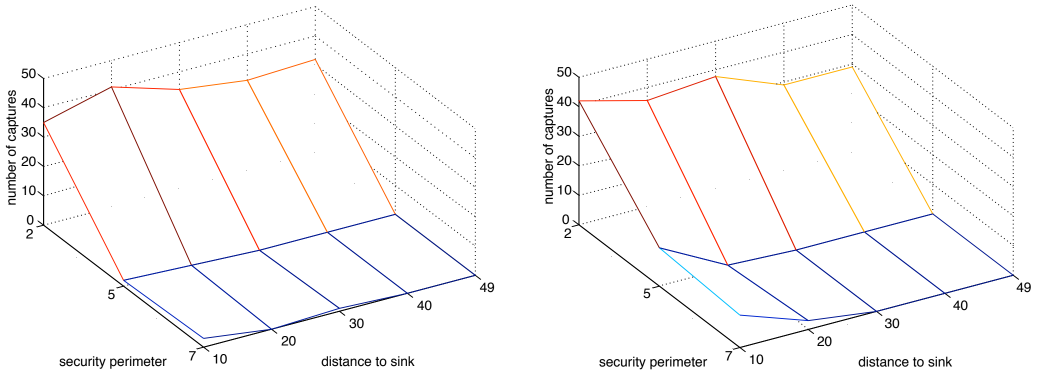
Figure 4.7: Path length distribution

4.4.4 Safety Distance Impact

In this section we study the impact of the security perimeter on the privacy protection level and the mean path length. Security perimeters of size 2, 5, and 7 have been used for the evaluation. Again, we have considered an inquisitive adversary and source nodes at various distances from the base station. The results on how the security perimeter affects privacy protection and the length of data paths are given in Figure 4.8 and Figure 4.9, respectively.

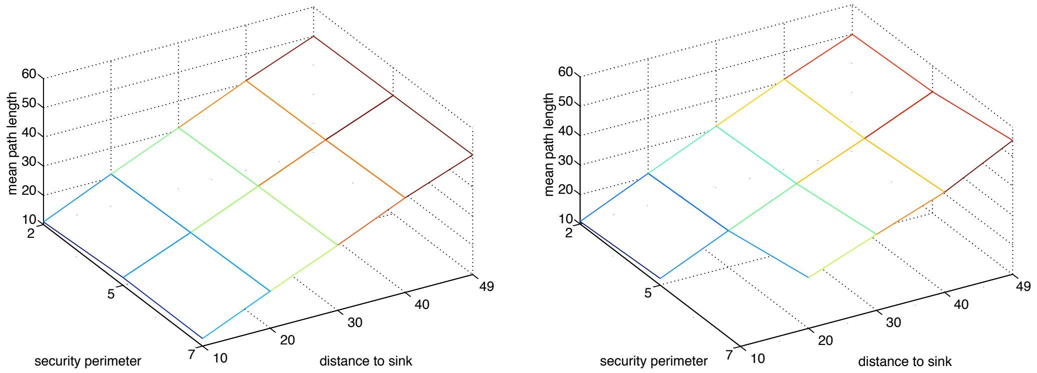
As expected, the size of the security perimeter has a clear impact on the number of captures. The larger the security perimeter, the better the privacy protection. In general, both the permissive and strict versions of the CALP mechanism behave well for a security perimeter size larger than 2. Also, the distance of the data source to the base station affects both versions but to a lesser extent. More precisely, the adversary is only able to capture a few packets in the permissive approach when the distance to the base station is not sufficiently large (see Figure 4.8a). This is also the case in the strict approach but the problem is even more acute (see Figure 4.8b). In general, the problem is that by using a small security perimeter the network is incapable of readjusting the routing paths and thus the adversary is more likely to capture packets, which leads him to the data source.

Additionally, we observe that the security perimeter size has an almost negligible impact on the mean path length. However, there might be some packets traversing an undue number of nodes before reaching the base station destination, as already discussed in Section 4.4.3. The strict version is more sensitive to



(a) Permissive CALP (b) Strict CALP

Figure 4.8: Impact on number of captures



(a) Permissive CALP (b) Strict CALP

Figure 4.9: Impact on mean path length

the size of the security perimeter. In particular, using a larger security perimeter when the source node is close to the base station might result in some executions with no packets reaching their destination. This particular case is depicted in Figure 4.9b for a security perimeter size of 7 and a source node located 10 hops away from the base station. Again, to counter the problem of having some packets not reach their target, a source-sink distance-dependent security perimeter might be used. In other words, the security perimeter might be larger for nodes that are located further away from the base station. Moreover, as the security perimeter size increases, the mean path length increase is more abrupt in the strict version than in the permissive version.

4.5 Summary

This chapter has presented the CALP scheme, a novel approach to source-location privacy that, unlike previous solutions, is triggered upon the detection of the adversary only. This mechanism benefits from the ability of sensor nodes to detect the presence of objects in their vicinity to prevent the transmission of messages in the area controlled by the adversary. When a sensor node detects the adversary it disseminates this information throughout the network thus enabling efficient privacy-preserving routing protocols.

The idea of feeding routing protocols with the location of the adversary has been successfully applied to a shortest-path routing technique. The combination of a shortest-path routing with the CALP scheme has a very clear advantage. Data packets only deviate from the most energy-efficient routing path when the adversary is in the proximity to that path. In particular, two versions of the protocol have been developed based on the way data packets are forwarded when an adversary is within a minimum safety distance from the sender. Moreover, two different strategies have been considered for the adversarial model. The extensive simulations that have been performed have demonstrated that the resulting protocol is capable of providing a solid privacy protection level with an average energy consumption very close to optimal.

Chapter 5

Probabilistic Receiver-Location Privacy

Wireless sensor networks are continually exposed to different types of attacks but the most devastating ones are those that target the base station since this critical device is responsible for collecting and analysing all the traffic generated in the network. Therefore, protecting the location of the base station 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 usually related to a highly-relevant facility. As a result, a number of authors have struggled to provide receiver-location privacy, primarily, by randomising and normalising the traffic pattern of the network. However, this might be insufficient when the adversary is also capable of retrieving the routing tables of the sensor nodes. Normally, the routing tables contain information regarding the distance to or the location of the base station, which may be used by the attacker to effectively reach the base station thus rendering useless anti-traffic analysis techniques. This serious threat to receiver-location privacy has never been taken into consideration in the literature.

This chapter presents HISP-NC (Homogeneous 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. This solution addresses, for the first time, both problems in a single solution. On the one hand, the 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, the 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 chapter is organised as follows. Section 5.1 describes the network and threat models as well as the main assumptions applicable to the rest of the chapter. A detailed description of the HISP-NC data transmission and routing tables perturbation schemes are presented in Section 5.2 and 5.3, respectively. Then, Section 5.4 evaluates and analyses the potential limitations of our solution with respect to the traffic overhead and delivery time of data packets. Finally, Section 5.5 presents a discussion and evaluates the privacy protection level achieved by the HISP-NC scheme under different types of attacks.

5.1 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.

5.1.1 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 [79] or a counter that is incremented for each transmitted packet.

5.1.2 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.

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.

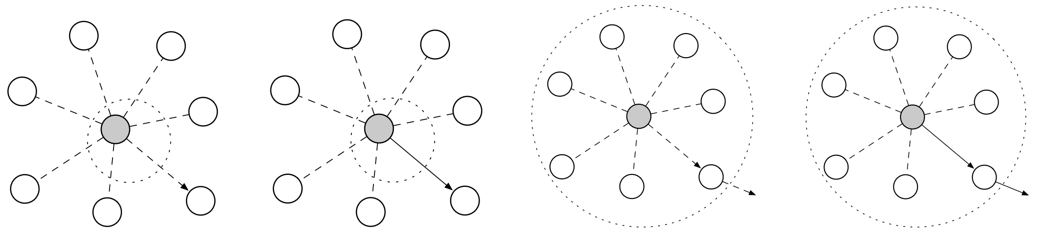
First, we consider the eavesdropping capabilities of the adversary. In particular, we concentrate on both the hearing range and the ability to retrieve packet header information. With respect to the hearing range, adversaries might range from those capable of observing the transmissions of a single node 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 [30]. Next, we provide a formal definition of the adversarial model:

Definition 3 ($\mathcal{ADV}$)**.** *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.*

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.



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

Figure 5.1: Adversarial Model Examples

The attacker model considered here has a limited 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 5.1. 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 [127]). 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 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

[1] The hearing range of current sensor nodes operating outdoors is around 100 meters for low power configurations [49]. However, these values might be altered by many factors such as the signal frequency or the presence of obstacles.

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.

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 [13] 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 5.2), 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., [32, 138]) 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 5.5.

5.2 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.

5.2.1 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.

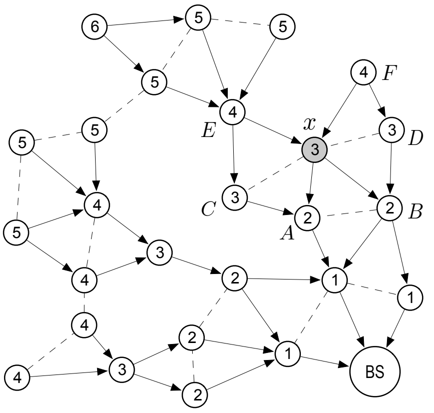
This process is guided by a computationally inexpensive approach that determines the recipients of messages. A node selects two neighbours by picking uniformly at random, a combination resulting from all the combinations of two elements without repetitions from its routing table. Since the routing tables 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.

5.2.2 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 5.2. In this figure we represent a particular network configuration and the routing table[2] of node x, which is three hops away 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.

[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 5.2: Routing table of shaded node x

5.2.3 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.

5.2.4 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 [68].

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., $[L^{\mathcal{C}}, L^{\mathcal{E}}, L^{\mathcal{F}}]$) 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 3 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

Algorithm 3 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
```

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.

5.3 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.

5.3.1 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.

5.3.2 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 5.2). 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.

5.3.3 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 4 (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 5.1, 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
$\ldots$		$\ldots$
1	→	e_6
0	→	f_5

Table 5.1: 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* $\sum_{n \in L^{\mathcal{C}}} pos(n) > \sum_{n \in L^{\mathcal{F}}} pos(n)$

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:

$$\mathbb{P}(n_1 \in L) = \frac{1}{C} \sum_{n \in L} pos(n) \tag{5.1}$$

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

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

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

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 5.1 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:

$$bias(r) = \frac{1}{C}(\sum_{n \in L^{\mathcal{C}}} pos(n) - \sum_{n \in L^{\mathcal{F}}} pos(n)) \tag{5.2}$$

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 5.1.

□

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 A.1 can be written in the following form:

$$bias_M(r) = \frac{1}{C}(\sum_{i=1}^{c}(N - i) - \sum_{i=1}^{f}(i - 1)) \tag{5.3}$$

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:

$$bias_m(r) = \frac{1}{C}(\sum_{i=1}^{c}(i-1) - \sum_{i=1}^{f}(N-i)) \tag{5.4}$$

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

$$\frac{c(c-1) - 2fN + f(f+1)}{N(N-1)} \leq bias(r) \leq \frac{2cN - c(c+1) - f(f-1)}{N(N-1)} \tag{5.5}$$

5.3.4 Perturbation Algorithm

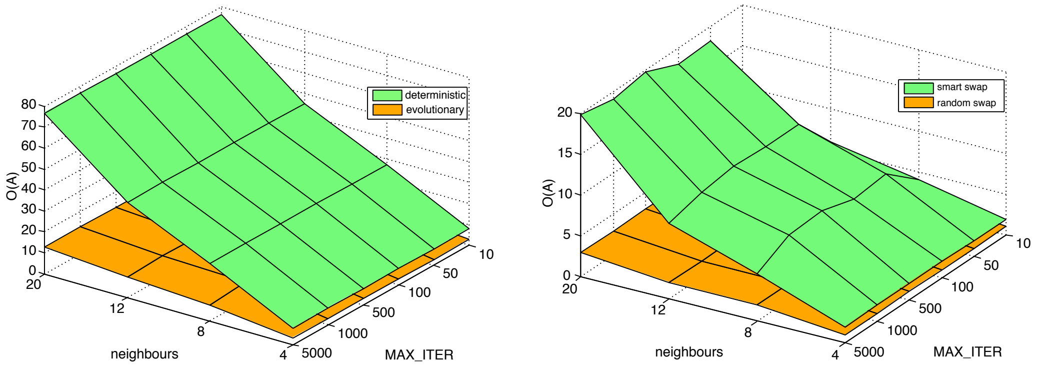
The perturbation algorithm in HISP-NC receives a routing table and a desired value and outputs an ordering of the table that satisfies, to some degree, the input bias value. This algorithm must be implemented by all nodes in the network and given the hardware limitations of the nodes, the complexity 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 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 [44] where simple mutations are applied to the routing table in order to minimise the distance to the desired bias. In Figure 5.3a we compare the order of complexity of a deterministic version of the algorithm versus



(a) Deterministic vs. Evolutionary

(b) Random swap vs. Smart swap

Figure 5.3: Complexity of perturbation algorithms

its evolutionary counterpart. Clearly, 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 8) 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 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.

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

Algorithm 4 Perturbation Algorithm

```
Input: $br \leftarrow \{L^{\mathcal{C}}, L^{\mathcal{E}}, L^{\mathcal{F}}\}$
Input: $bias$, $MAX\_ITER$
$E \leftarrow energy(bias, br)$
$i \leftarrow 0$
while $(i < MAX\_ITER) \wedge (E \neq 0)$ do
    $br' \leftarrow swap(br)$
    $E' \leftarrow energy(bias, br')$
    if $(E' < E)$ then
        $br \leftarrow br'$
        $E \leftarrow E'$
    end if
    $i \leftarrow i + 1$
end while
return $br$
```

Figure 5.3b 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 5.5), 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.

5.4 Protocol Analysis

This section presents a detailed analysis on the potential limitations that might hinder the successful operation of the HISP-NC protocol. 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.

5.4.1 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 5.3.3, 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:

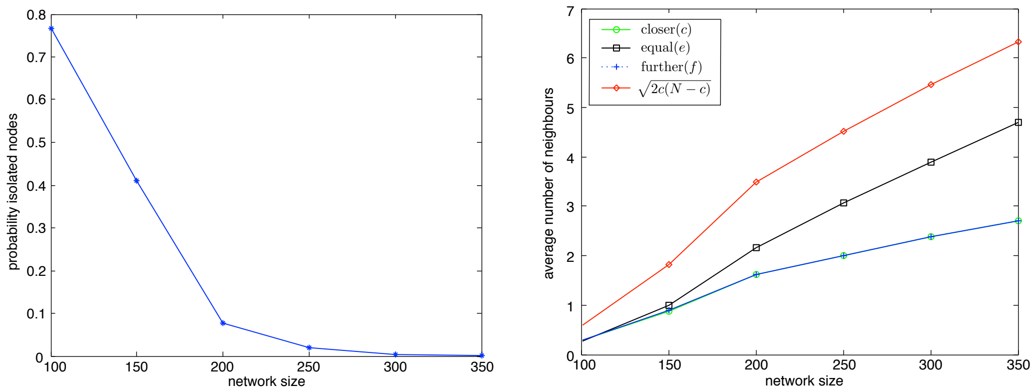
$$f < \sqrt{2c(N-c)} \tag{5.6}$$

Therefore, we might say that if Equation 5.6 is satisfied, then the following implication holds:

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

□

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 5.4 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 5.4a we show that the probability of isolated nodes drops significantly when the network size is over 200 nodes. Moreover, Figure 5.4b presents the average number of neighbours closer, equal and farther for any node in the network. In



(a) Probability of isolated nodes

(b) Neighbours restriction

Figure 5.4: Node connectivity in randomly deployed networks

this figure we also show that the restriction imposed by Equation (5.6) on the maximum number of further neighbours is satisfied at all times.

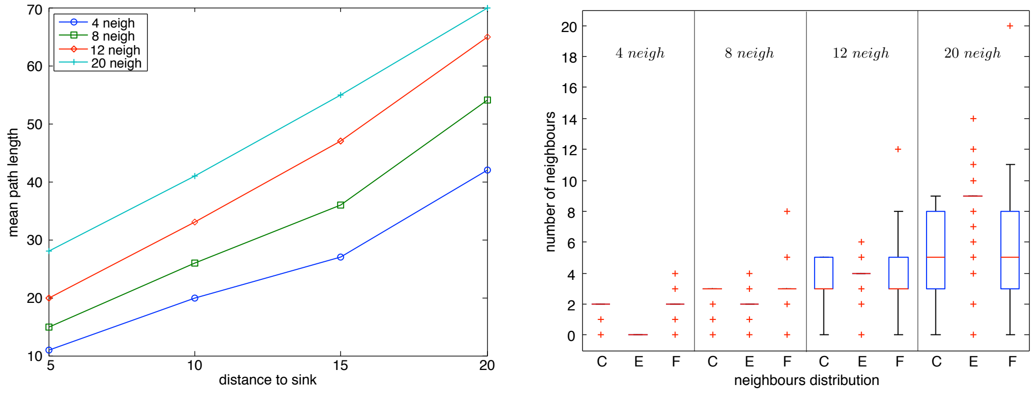
Note that the results shown in Figure 5.4b are average values and there might be some nodes not satisfying the restriction. However, this would only pose some additional delay unless there are network regions with a high concentration of nodes unable to fulfil the imposed condition. This issue might 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.

5.4.2 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:



(a) Expected number of hops

(b) Distribution of neighbours

Figure 5.5: Protocol performace for various network configurations

$$x_n = 1 + px_{n-1} + qx_n + rx_{n+1} \tag{5.7}$$

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 5.5 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.

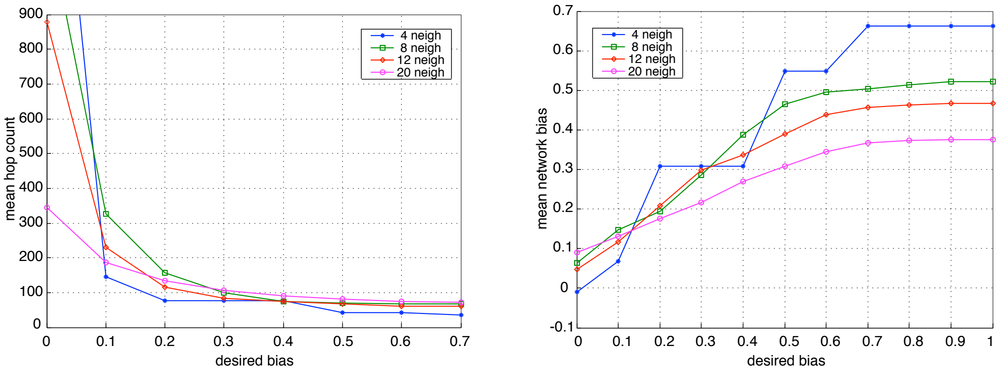
The results show that the expected number of hops increases with the distance to the sink as well as with the connectivity of the nodes. As the number of neighbours available to a node increases, the more difficult it is for the adversary to make a decision on which of the recipients is actually closer 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 5.5b, 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 5.5a 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 the transmission range of the nodes, being more significant for the $20neigh$ 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 5.6a, where the mean number of hops travelled by packets is depicted, and Figure 5.6b, 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 5.6b we can observe that for those configurations with a larger



(a) Mean number of hops (b) Mean network bias

Figure 5.6: Perturbation impact on message delivery

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 $4neigh$ 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 $20neigh$ 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.

5.4.3 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.

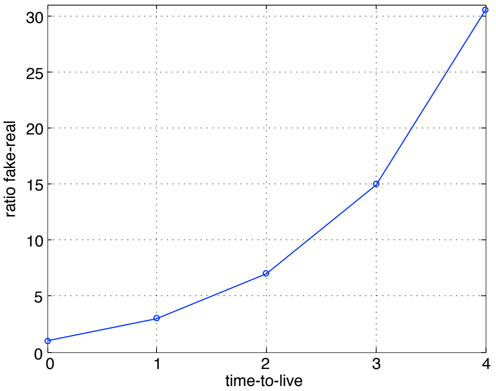


Figure 5.7: Overhead of fake messages

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 5.7 we illustrate the overhead imposed by HISP-NC for different time-to-live values. 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. Note that 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.

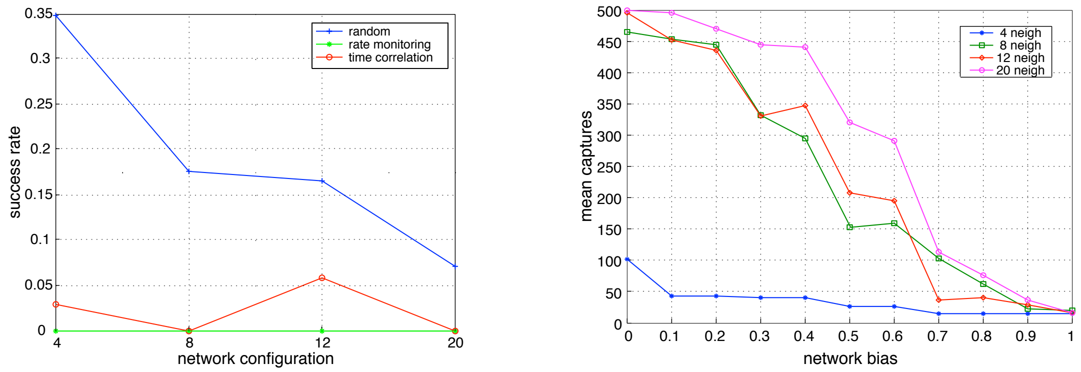
Finally, note the overhead imposed by fake transmissions might be reduced by half if we introduce a slight modification. Instead of sending two packets upon the reception of traffic, we might send a single packet addressed to two node identifiers. In this way, and assuming that the identifiers are hidden 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.

5.5 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



(a) Traffic Analysis Attacks

(b) Node Capture Attacks

Figure 5.8: Success rate of different adversaries

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 5.4. 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 5.8a.

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 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

[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.

(see Figure 5.8b). 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 (perturbed) 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.

5.6 Summary

This chapter has presented HISP-NC, a receiver-location privacy solution that aims to prevent both traffic analysis and routing table inspection attacks. The solution consists of a traffic normalisation scheme, which relies on the injection of controlled amounts of fake traffic to hide the flow of real traffic, and a routing table perturbation scheme, which reorders the elements of the routing table according

[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.

to a given bias in order 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. We have also 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 eavesdroppers and attackers capable of inspecting the routing tables of a limited number of nodes in the network.

Chapter 6

Conclusion

This chapter recapitulates our efforts to accomplish the objective of protecting one of the most critical pieces of contextual information in WSNs, namely, location information. We start by briefly summarising the research scope and problems that have been addressed in this thesis. We then present a short description of each of our contributions to the field while expanding on some possible lines of improvement. Finally, we introduce open problems that need further research to facilitate the adoption of this technology in everyday scenarios.

6.1 Contributions

Wireless sensor networks provide a distributed and self-managed sensory system for monitoring and interacting with the physical world from remote locations. The information collected by these systems is very diverse in nature due to the number of sensors that can be fitted into these wireless-enabled, matchbox-sized computers. This versatility allows the seamless integration of sensor nodes in any conceivable scenario, being especially useful in remote and hostile environments but they are also suitable for industrial control, precision agriculture, habitat monitoring, and so forth.

This technology brings tremendous benefits to both businesses and individuals but it also poses serious privacy risks due to the myriad data they are capable of collecting unnoticed. This data collection problem has long been acknowledged by the research community, who have struggled to devise innovative solutions to it. Besides this unsolicited surveillance, there are some additional privacy issues that arise from the wireless nature of the communications in WSNs. These new privacy problems are due to the analysis of the metadata associated with the measurement and transmission of the data collected by the sensor nodes and it primarily affects the network infrastructure and the phenomena being monitored by the network, which might be directly related to individuals, goods, or business processes. Therefore, contextual information privacy becomes an essential service for wireless sensor networks.

A noteworthy contextual privacy problem is location privacy. Attackers can determine the location of both the nodes reporting data and the base station, by observing the data flows in the network. Therefore, the general approach to protecting location privacy in sensor networks is to obfuscate the traffic pattern as done in traditional anonymous communication systems originally devised for computer-based communications. Therefore, the first step towards achieving the final goal of this thesis was to analyse and understand whether computer-based anonymity systems could be adapted to solve the location privacy problem in sensor networks. To that end, we studied which anonymity properties were most suitable to keep the adversary away from the data sources and the base station. We concluded that in order to be able to deal with external attackers it is necessary to hide the presence of real packets and thus undetectability and unobservability are desired properties. In contrast, other properties like anonymity and unlinkability are counterproductive or unnecessary in event-driven sensor networks. Additionally, we wished to study the features and techniques used by computer-based anonymity systems to asses whether some of them could be applied to sensor networks. Both centralised and decentralised anonymity solutions were analysed and we concluded that decentralised solutions are generally more suitable for sensor networks because all the nodes behave similarly, hindering traffic analysis. We also paid particular attention to the overhead imposed by these solutions to determine whether some of them could fit the constrained hardware of sensor nodes. Some of these solutions are lightweight enough but were devised with a different goal and adversarial model in mind, while other solutions fit the requirements of the sensor domain but are either overly costly or they limit the functionality of the network. In essence, the main conclusion of this research is that, despite their apparent similarity, computer-based anonymous communication systems are not suitable for the particular requirements and adversarial models being considered, therefore it became obvious the need to devise new tailored solutions for WSNs.

This led us to review and categorise the existing literature on privacy in WSN. We surveyed more than 50 location privacy solutions and grouped them into three main categories, namely, identity protection, source protection, and receiver protection. These groups present different sub-categories based on the main features of the solution and the capabilities of the adversary to be countered by them (see Figure 3.22). We also analysed their pros and cons, discovered open problems, and identified research gaps to contribute to. One of the biggest problems that was identified when reviewing the literature is that location privacy solutions usually imply larger communication paths, which in turn results in increased energy consumption, a higher probability of packet loss, and larger delivery delays. This raised the following research question: Is it possible to activate the privacy protection mechanisms only when necessary? This is precisely the idea behind the CALP scheme, our context-aware location privacy solution. It benefits from the ability of sensor nodes to detect objects in their vicinity so as to determine the location of an attacker in the field. Using this information, the network can dynamically readapt the routing paths in order to circumvent the area where the adversary is located. The scheme was successfully combined with a single-path routing algorithm to provide source-location privacy with minimal overhead when the adversary is not present in the field. Specifically, we have devised a strict and a permissive CALP version of the scheme, both of which provide a solid privacy protection. The main difference between them lies on the penalties imposed on the routing through the area controlled by the adversary. In the strict version, the penalties are so high that packets do not traverse this area while in the permissive version, this is possible. The former is more secure but it may impose overly high transmission delays if the adversary remains static next to the base station. Therefore, we concluded that switching from a strict to a permissive approach as packets approach to the base station is the most suitable approach to protect from both inquisitive and patient adversaries. Although the CALP mechanism has proven to be efficient in the context of source-location privacy we believe that it is also suitable for protecting the location of the base station since it is based on the concept of unobservability.

In the area of receiver-location privacy, we observed that existing solutions targeting local adversaries were either too costly or leaked location information in certain circumstances. Moreover, we realised that, unlike solutions for source-location privacy, no scheme considered the threat of internal adversaries and routing tables inspection. This is a serious threat to receiver-location privacy affecting all of the solutions analysed in this thesis since the routing tables of the nodes contain relevant information about the location or the direction to the base station. This information is necessary for the routing protocols to deliver data packets to the destination but it can be exploited by an attacker to effectively reach the base station, thus rendering the efforts made by the network in deploying anti-traffic analysis mechanisms absolutely useless. Our solution, HISP-NC, provides robust probabilistic protection against local adversaries capable of performing typical traffic analysis attacks but also, and for the first time, routing table inspection attacks. During data transmission, the devised solution sends data packets on a biased random walk towards the base station and hides this information flow with controlled amounts of fake traffic. In this way, the overall number of packets a node distributes between its neighbours is homogeneous and thus statistically safe while ensuring the convergence of real traffic to the base station. Additionally, the HISP-NC scheme provides a routing table perturbation scheme that rearranges the elements of the routing table in order to reduce the risk of node compromise attacks. The perturbation scheme is based on the concept of routing table bias, which defines the speed at which packets converge on the base station and depends on the ordering of the table. The solution allows the network operator to indicate a desired bias, which is useful to reach a balance between privacy protection and data delivery time.

6.2 Challenges and Future Work

Privacy preservation in WSNs has proved to be an extremely challenging task and regardless of the number of schemes that have been devised there are still challenges and open problems that remain unresolved and demand further research and innovative solutions. This thesis has covered just a subset of privacy problems within the area of contextual information privacy and there is still room for improvement. Probably, the most obvious lines of research in the short term are the enhancement of the proposed source- and receiver-location privacy solutions.

First, we wish to develop and evaluate the apparent benefits of combining a strict and permissive CALP approach. We are also willing to extend the CALP scheme to provide a holistic location privacy solution that is able to preserve the location of both data sources and the base station. A naive solution to these problems is to use baseline flooding together with fake data sources but this approach is too energy consuming for battery-powered devices. Achieving an effective and energy-efficient solution would be a significant breakthrough in the area since there is yet to be a single scheme capable of providing an integral solution to both problems simultaneously. The CALP scheme seems promising in this respect because it offers unobservability at a very low cost by modifying the data communication paths based on the location of the adversary. To that end, we need to further investigate the way in which to provide sensor nodes with the ability to precisely identify and trace adversaries. By monitoring the movements of the adversary and not only the current position, the network may be able to infer the strategy of the attacker as well as its target. Moreover, we are interested in taking into account the threat of internal adversaries. The proposed solution assumes that route update messages are secure and legitimate. But what if we cannot presume that? what if messages are legitimate but they are being created by compromised nodes? We will look into the notions of reputation and trust as they seem suitable to identify and revoke nodes that misbehave during the route updating process.

The receiver-location privacy solution that has been devised in this thesis is robust against existing traffic analysis attacks but its overhead increases exponentially with the hearing range of the adversary since fake traffic needs to extend beyond the area he controls. Therefore, we are currently concentrating our research efforts on reducing the amount of fake traffic necessary to maintain an adequate protection level for the base station against powerful adversaries. The protection against external adversaries has focused on the data transmission phase but the location of the base station can also be leaked during the network topology phase. The network is regularly flooded with a hop count message that allows each sensor node to determine the distance of its neighbours from the base station. This process is initiated by the base station, thus revealing its location to external observers. We are working to incorporate an anonymous topology discovery protocol in the HISP-NC scheme in order to provide a complete solution to receiver-location privacy in WSNs.

Besides the aforementioned improvements, we wish to look into other challenging open problems. The literature lacks an interoperability framework that allows researchers to quantify and compare the location privacy protection level of solutions. Currently, different authors resort to different approaches such as measuring entropy, game theory, evidence theory, numerical analysis, and simulations. However, it is not trivial to provide a formal model that accurately represents the behaviour of the system, especially in the context of a local adversary. Although it is possible to measure the privacy loss in one step, the information leak accumulates in such a way that it remains intractable as the adversary moves through the field. Probably, this is the reason why simulations is the most common approach to proving the correctness of solutions. But simulation results are not easily reproducible because either the simulator is not standardised or the code is not made publicly available, or both. Thus, defining an interoperability framework is a challenging area of research that may help to devise enhanced solutions.

In line with the previous issue, it is necessary to formally and faithfully define the capabilities and actions that may be performed by the adversary. The traditional approach is to define an adversary with a predefined strategy that remains unaltered. An appropriate model for representing the knowledge of the adversary does not exist. At most, the adversary knows whether or not he has already visited a specific node. The adversary does not use or infer new information based on previously known data or additional sources of information. In this regard, the adversarial model considered in the literature is mostly passive and does not interfere with the normal operation of the network. Particular attention must be paid to adversaries who can inject, modify, reply, or block messages from a portion of the network given the hardware limitations of sensor nodes. Also, more research must be conducted to devise solutions against internal adversaries, which are not only capable of obtaining contextual information but also packet contents.

This thesis has concentrated on a specific wireless sensor network scenario where sensors nodes are static and they communicate solely with the base station. However, one of the most promising areas for research and innovation, the Internet of Things, opens the door to new scenarios where everyday objects are fitted with sensors, actuators, and limited batteries, just like sensor nodes. In this setting, mobility is of paramount importance as devices may be carried or moved to different locations and they might also be directly associated with individuals. Moreover, these computing devices will not only interact with a single base station but also with other near or remote devices through the Internet. Consequently, the solutions that have so far been devised are no longer useful. Similarly, new types of adversaries might appear. Therefore, we believe that the integration of sensor networks with the Internet will result in a prolific area of study.

Appendices

Appendix A

Resumen en español

A.1 Redes de sensores y privacidad

Las redes inalámbricas de sensores (Wireless Sensor Networks [49]) son sistemas de monitorización altamente distribuidos compuestos por nodos sensores y una o más estaciones base. Los nodos sensores son dispositivos de capacidad computacional y tamaño reducido, capaces de sentir fenómenos físicos en su entorno y de comunicarse de manera inalámbrica. La estación base es un dispositivo de mayor capacidad que se encarga de recopilar, procesar y ofrecer la información obtenida por los nodos sensores a los usuarios de la red.

Gracias a su versatilidad y reducido tamaño, estas redes han demostrado ser una tecnología ideal para la monitorización y control de infinidad de escenarios. Sin embargo, su reducido tamaño y el hecho de que los nodos sensores se encuentren alimentados por pilas o baterías es también uno de sus mayores inconvenientes, ya que limitan su capacidad de cómputo, sus posibilidades de almacenamiento y su tiempo de vida. Esto influye de manera trasversal en el desarrollo de protocolos y aplicaciones, ya que se hace obligatorio el uso responsable de los limitados recursos disponibles. Asimismo, esto afecta de manera notable a la seguridad de estos sistemas, que se convierten en el blanco de diferentes tipos de amenazas y ataques [139].

A pesar de la infinidad de trabajos dedicados a la protección de las redes de sensores en todos sus niveles, tanto hardware como software, la protección de la privacidad no ha recibido la suficiente atención de la comunidad científica. En general, podemos distinguir dos tipos de problemas de privacidad derivados del despliegue de este tipo de redes. El primero y más natural se debe a la capacidad que tienen las redes de sensores de pasar desapercibidas al tiempo que recopilan y correlacionan información acerca de los individuos o entidades que se encuentren dentro de su ámbito de acción [145]. Este tipo de amenaza no puede ser afrontada únicamente con medios tecnológicos sino que además requiere de legislación, auditorías y sanciones severas para persuadir a posibles infractores. El segundo tipo de problema, que es en el que se centra la presente tesis, afecta a la privacidad de la propia red de sensores y, en consecuencia, también puede afectar a las entidades y objetos que ésta monitoriza. En este ámbito encontramos dos categorías en función de si las soluciones se centran en proteger el contenido de los paquetes o el contexto en el que se desarrolla la actividad de la red. Así pues, tenemos *content-oriented* y *context-oriented privacy* [100].

Una primera línea de defensa para proteger la privacidad de la red es aplicar esquemas de cifrado seguros que permitan preservar la confidencialidad de los datos transmitidos. Esta medida garantiza que entidades externas no tengan acceso al contenido de los paquetes. Sin embargo, esto por si sólo no es suficiente para garantizar la privacidad ya que estos datos estarían disponibles a posibles atacantes internos (nodos legítimos controlados por un adversario). Asimismo, un observador externo podría inferir información sensible a través de los atributos de la comunicación. De hecho, la mera presencia de paquetes puede revelar información a un atacante. Por ejemplo, en una red dedicada a monitorizar el trasiego de individuos en un edificio, la presencia de paquetes significaría que se ha detectado una persona en un área determinada, independientemente del contenido del mensaje. Este es un claro problema de privacidad.

A.1.1 Privacidad de localización

La privacidad de localización puede definirse como el deseo de decidir en qué casos y con qué precisión se expone información de localización a terceras partes [5]. En una red de sensores, dependiendo de la entidad cuya localización queramos proteger podemos tener privacidad de origen y privacidad de destino. La privacidad de fuente o *source-location privacy* pretende mantener oculta la localización de los nodos que reportan eventos. El objetivo final no es la protección de los dispositivos sino garantizar que un atacante no puede determinar el área donde tienen lugar determinados eventos, ya que los eventos pueden estar asociados a individuos o recursos de gran valor económico o estratégico, como ocurre en el escenario descrito en la Figura A.1. Por otra parte, la privacidad de destino o *receiver-location privacy* tiene que ver con la protección de la estación base. Este dispositivo es de vital importancia para la integridad y supervivencia de la red porque si se viera comprometido o fuera destruido, todo el sistema dejaría de ser de utilidad. Pero además de para garantizar la protección física de la red, la ubicación de la estación base es sensible porque suele alojarse en una instalación de gran relevancia. Por ejemplo, en el escenario de la figura, la estación base se encuentra en el campamento militar, por lo que al conocer su ubicación el atacante obtiene una gran ventaja sobre su enemigo.

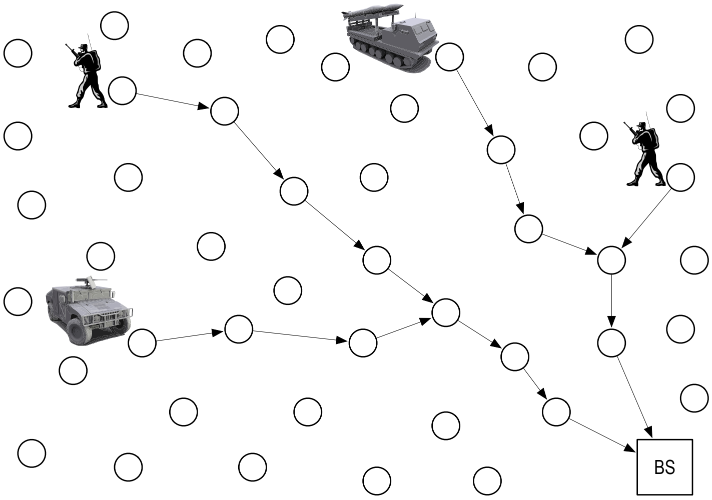


Figure A.1: Red de sensores desplagada con fines militares

Ambos problemas se deben al modelo de comunicación particular de las redes de sensores. Estas redes suelen desplegarse con el fin de ofrecer un sistema de monitorización en tiempo real. Por ello, tras la detección de un evento de interés (p. ej., la presencia de tropas) en las inmediaciones de un nodo sensor, éste reporta inmediatamente a la estación base utilizando un protocolo de comunicaciones multi-salto. Por lo general, con el fin de ahorrar energía, se hace uso de protocolos que buscan el camino óptimo. El uso de este tipo de protocolos da lugar a marcados patrones de tráfico, que facilitan a un atacante localizar el origen y destino de las comunicaciones.

El atacante suele considerarse pasivo y externo. Un atacante externo es aquel que no tiene control sobre la infraestructura y por tanto no tiene acceso a los secretos compartidos por la red u otra información. Se dice que un atacante es pasivo cuando no interfiere con el comportamiento normal de la red, es decir, se limita a observar el comportamiento. En función de su capacidad de observación se distinguen dos modelos de atacante, los atacantes locales y los globales. Típicamente, los atacantes locales tienen un rango de escucha similar al de un nodo sensor y, por tanto, suelen moverse por el área de despliegue de la red siguiendo paquetes con el fin de alcanzar su objetivo. La estrategia para determinar su siguiente movimiento dependerá de si su objetivo es alcanzar nodos fuente de evento o la estación base. En cambio, los atacantes globales tienen un rango de escucha mucho más amplio, equivalente a toda la red, que suele conseguirse mediante el despliegue de una red de sensores propia que monitoriza la tasa y tiempos de envío de la red de sensores legítima.

Cuando el objetivo es localizar a los nodos origen, la estrategia seguida por un atacante local es realizar un ataque conocido como *traceback attack*. Para ello, el atacante cuenta con una antena direccional capaz de estimar el ángulo de llegada de los paquetes que observa y, con esta información, puede moverse hacia el nodo que realizó el envío. De esta forma el atacante va reduciendo, salto a salto, su distancia al nodo origen. En cambio, cuando el objetivo es encontrar la estación base, el atacante local opta por monitorizar los tiempos de envío entre nodos vecinos o por observar la tasa de envío de los nodos a su alrededor. La primera estrategia, conocida como *time-correlation attack*, permite al atacante determinar la dirección hacia la estación base gracias al hecho de que, cuando un nodo recibe un mensaje, lo reenvía inmediatamente hacia su destino. Así pues, el atacante puede saber qué vecino está más cerca de la estación base al observar que éste retransmitió el mensaje. La segunda estrategia, conocida como *rate-monitoring attack*, se basa en el hecho de que los nodos más próximos a la estación base tienen una tasa de envío mayor al tener que enviar no sólo su propio tráfico sino también el de nodos remotos. Por ello, el atacante se mueve sucesivamente hacia el nodo con la tasa de transferencia más elevada de su entorno.

A.2 Adecuación de los sistemas de comunicación anónima

Anteriormente hemos establecido que el problema de la privacidad de localización se debe a los marcados patrones de tráfico característicos de las redes de sensores, que permite a posibles observadores determinar el origen y destino de las comunicaciones. Dado que los sistemas de comunicación anónima (ACS) para redes de ordenadores fueron diseñados con el fin de dificultar el análisis de tráfico, estos pueden ser una solución plausible al problema. Sin embargo, la literatura especializada [94, 140] se ha limitado a desechar el amplio espectro de soluciones existentes con argumentos demasiado vagos, que se centran en la limitación de recursos de los sensores. Consideramos que este argumento no es suficiente y cometeríamos un serio error al descartar estas soluciones sin un análisis pormenorizado, ya que en el futuro las capacidades de los sensores pueden mejorar y equipararse a las ofrecidas por los equipos de sobremesa actuales.

A.2.1 Propiedades de anonimato en WSN

Si bien una de las características fundamentales de los ACS es que proporcionan mecanismos para dificultar el análisis de tráfico, no todos estos sistemas persiguen las mismas propiedades de anonimato. Del mismo modo, como veremos a continuación, no todas las propiedades son de utilidad en redes de sensores.

El *anonimato* es la capacidad que tiene un individuo de no ser suficientemente identificable entre un grupo de sujetos con los mismos atributos. Por lo general, lo que persigue un sistema de anonimato es preservar la identidad de las partes que intervienen en una comunicación, es decir, la identidad del emisor y la del receptor. En el ámbito de las WSN la utilidad de esta propiedad está limitada a ciertas situaciones, llegando a ser contraproducente en otras. Dado que la estación base necesita saber en todo momento la identidad del nodo que envía la información para una correcta gestión de los eventos, si se proporciona anonimato a los nodos origen, la red dejaría de ser de utilidad. Asimismo, los nodos deben conocer el identificador de la estación base para poder enviarle la información recopilada. No obstante, esta propiedad es interesante para hacer frente a atacantes internos y observadores externos que poseen un mapa de la red. Por tanto, el anonimato sólo es de interés en ciertas ocasiones y ante determinadas entidades.

De especial importancia para las comunicaciones tradicionales es la propiedad de no enlazabilidad o *unlinkability*. Esta propiedad asegura que un atacante no es capaz de distinguir fehacientemente si dos o más objetos de interés están relacionados. Los ACS suelen esforzarse por proporcionar no enlazabilidad entre fuente y destino, ya que sin esta propiedad, un atacante puede hacer perfiles de usuarios en función de los sitios que visita. Esta idea no tiene demasiado sentido en redes de sensores convencionales ya que el flujo de comunicación apunta siempre a la estación base. En estas redes, la enlazabilidad es un problema si el atacante es capaz de determinar que un paquete pertenece a un nodo determinado ya que esto le guiaría directamente a la zona de la red donde se produce el evento. En tal caso, estamos ante la misma situación que presentamos anteriormente para la propiedad de anonimato.

Por último, hay dos propiedades que se centran en la protección de los objetos de interés por sí mismos. La indetectabilidad o *undetectability* evita que un atacante pueda tener la certeza de que un objeto de interés existe. Por otra parte, la inobservabilidad o *unobservability* proporciona además anonimato a las entidades relacionadas con el objeto de interés. Por tanto, la indetectabilidad oculta la existencia de mensajes reales mientras que la inobservabilidad además implica que si los mensajes son descubiertos, sus emisores y receptores no pueden ser identificados. Estas propiedades son las más naturales para proteger la localización en redes de sensores. Si el atacante es incapaz de detectar la presencia de mensajes de evento, tampoco podrá determinar la localización de los nodos que se comunican.

A.2.2 Análisis de soluciones tradicionales

Las propiedades antes mencionadas han sido satisfechas por los sistemas ACS a través de diferentes técnicas, con mayor o menor impacto en las comunicaciones y la carga computacional de los sistemas. Estas técnicas van desde un simple cambio de identidad hasta operaciones más complejas y costosas, tales como la aplicación de cifrados sucesivos, la inyección de tráfico falso, y comunicaciones sincronizadas. Por otra parte, los sistemas de comunicación anónima se pueden clasificar en centralizados o distribuidos, dependiendo de si los usuarios colaboran en el proceso de anonimización. A continuación analizaremos varios sistemas ACS con el fin de determinar si tanto las técnicas utilizadas como el consumo de recursos se adecuan a las características y necesidades de las WSNs.

Sistemas centralizados

Las soluciones *single-proxy* [10] consisten en un único dispositivo que hace de intermediario en una comunicación, de manera que cuando el emisor manda un mensaje al destinatario, en lugar de hacerlo directamente, lo hace a través del proxy, que cambia el identificador del paquete por el suyo propio, ocultando así la identidad del emisor original. Desde un punto de vista computacional, este tipo de soluciones impone un coste computacional mínimo, siendo el proxy el que mayor carga de trabajo realiza, más aún si tenemos en cuenta que el proxy suele dar servicio a múltiples clientes. Sin embargo, este tipo de soluciones por si solas sólo ocultan la identidad del emisor pero son incapaces de evitar los ataques de tráfico típicos en redes de sensores. Cuando el atacante es local, éste es capaz de alcanzar su objetivo porque los paquetes siguen siempre la misma ruta. En el caso de que el atacante global, es trivial determinar los extremos de la comunicación, al igual que pasa con todas las soluciones centralizadas. Sin embargo, el renombrado de paquetes puede proporcionar cierto nivel de protección frente a atacantes internos, al menos para aquellos que se encuentran detrás del proxy.

Las redes de mixes o *mix-nets* [26] están formadas por un conjunto de mixes (i.e., proxies) que están ideadas para comunicaciones tolerantes a retrasos. Cuando un usuario desea comunicarse con otro, selecciona una serie de mixes y, por cada uno de ellos, añade a su mensaje una capa de cifrado asimétrico en orden inverso. De esta forma, cada mix del camino sólo tiene acceso a su capa de cifrado y no puede obtener ni el contenido del mensaje ni el resto de nodos del camino, salvo su antecesor y sucesor. Además, como cada mix almacena todos los mensajes que recibe durante un periodo de tiempo considerable, se hace extremadamente difícil correlacionar los paquetes que envía y recibe. Debido a este importante retraso, su uso no es adecuado para la mayoría de WSNs, que requieren de capacidad de monitorización en tiempo real. Por otra parte, hay otras limitaciones con respecto a los requisitos computacionales y de memoria. En primer lugar, los nodos fuente tienen que crear tantas capas de cifrado asimétrico como mixes atraviese cada uno de sus mensajes. Esto supone, además, que los nodos deben conocer la topología de la red para ser capaces de aplicar las capas de cifrado en el orden correcto. Y deberán almacenar un gran número de paquetes durante un largo periodo de tiempo. Aunque este esquema es muy adecuado para hacer frente a atacantes internos, es incapaz de hacer frente a atacantes globales o atacantes locales que se muevan en el terreno. Estos últimos podrían llegar al extremo de la mix-net y seguir los paquetes ya que seguirían rutas fijas.

El esquema de *onion routing* [108] es muy similar al anterior ya que consta de una serie de proxies, conocidos como onion routers, pero su principal mecanismo de defensa no se basa en el retraso de paquetes sino en la ocultación de los caminos que estos atraviesan. Cuando un emisor necesita enviar información, primero establece una ruta o circuito dentro del sistema. El circuito se establece mediante una estructura de datos formada por varias capas de criptografía asimétrica que contienen las claves que cada intermediario usará para descifrar el flujo de datos posterior. Una vez establecido el circuito, el origen cifra repetidamente los datos con las claves simétricas establecidas con los miembros del camino. Al tener un funcionamiento similar a las mix-nets, su requisitos y limitaciones son parecidos. La diferencia principal radica en que no imponen un gran retraso en las comunicaciones pero sigue siendo necesario el cifrado (simétrico) por capas en origen y el conocimiento de la topología de la red para realizarlo en el orden correcto. Cada nodo intermedio descifra una capa de cada paquete recibido y los multiplexa en los enlaces cifrados que mantiene con otros miembros de la red. A pesar de que reduce el coste impuesto por las redes de mixes, sigue siendo un esquema demasiado pesado y además presenta las mismas limitaciones frente los tres modelos de atacante considerados.

Sistemas distribuidos

En el sistema *Crowds* [109], su propios miembros forman el sistema de anonimato, colaborando para enviar peticiones entre todos, ocultando así la identidad del verdadero origen. Cuando un miembro del sistema quiere enviar un mensaje, se elige otro miembro al azar para actuar como intermediario. El receptor decide si repetir el proceso o si enviar finalmente el mensaje al destino. Además, los nodos del camino descifran el mensaje, reemplazan la identidad por la suya propia y lo vuelven a cifrar para cambiar su apariencia. Los mensajes posteriores con mismo origen y destino seguirán el mismo camino. Aunque no se trate de un esquema con un coste computacional elevado, los requisitos de memoria son importantes. Cada nodo debe compartir una clave con cada miembro del Crowd y mantener una tabla que le indique a qué camino corresponde cada paquete recibido. Sin embargo, su mayor limitación es que al ser caminos estáticos pueden ser seguidos fácilmente por atacantes locales. A pesar de ser una solución descentralizada, los atacantes globales siguen siendo capaces de detectar el origen y destino de datos ya que no se ofrece ningún mecanismo para ocultar el envío y todo el tráfico generado acaba en la estación base. Sin embargo, los mecanismos utilizados pueden hacer frente a atacantes internos ya que sólo conocen el paso anterior y el siguiente de cada camino.

El protocolo *GAP* [14] fue ideado para permitir la compartición de archivos en redes P2P. La idea de base es que cuanto más tráfico transmite un nodo menos probable es que un mensaje particular haya sido generado por él. Por ello, la red mantiene una base de ruido, en forma de tráfico falso, que permite a los nodos empezar a transmitir sin exponerse a posibles atacantes. Cuando un nodo recibe un mensaje, éste puede decidir reenviarlo a varios miembros, reenviarlo tras reemplazar su identificador de origen o simplemente descartarlo. Además, los datos que atraviesan la red utilizan un esquema de codificación similar a un cifrado simétrico que permite a los nodos verificar si los paquetes que reciben concuerdan con alguna de sus peticiones sin necesidad de descifrar. Finalmente, se añaden pequeños retrasos para dificultar la correlación de tráfico. Este esquema impone un elevado coste, sobre todo a nivel de consumo energético. Por otra parte, el modelo de comunicación P2P no concuerda con el de una red de sensores, aunque la estación base podría comportarse como un miembro más. En cuanto al nivel de protección frente atacantes, este esquema es bastante robusto frente a observadores y atacantes internos. Tanto atacantes locales como globales son distraídos gracias al tráfico falso que se usa de sustento. Los atacantes internos son evitados en cierta medida gracias al renombrado de las cabeceras y el esquema de codificación de datos.

El modelo *DC-nets* [27] permite compartir información entre un grupo de usuarios al tiempo que ocultan al emisor (y destino), incluso frente al resto de participantes del protocolo. Cada ronda del protocolo permite la transmisión de un bit de información, para lo cual es necesario que cada nodo comparta un bit secreto con sus vecinos. Todos los miembros transmiten simultáneamente el resultado de sumar los bits secretos que conoce y si alguien tiene información que compartir invierte el resultado de la operación. Como cada secreto se utiliza dos veces, el resultado de sumar todas las contribuciones debe ser cero, a menos que alguien haya invertido su resultado. Dado que los bits compartidos son secretos, no hay manera de determinar el emisor. Este esquema puede extenderse para permitir la transmisión mensajes mediante la comparición de números aleatorios en lugar de bits. La aplicación del modelo DC-nets en redes de sensores se encuentra limitada por varios factores. En primer lugar, la necesidad de un canal de emisión sincronizado y fiable que cubra a todos los nodos de la red, incluida la estación base. En segundo lugar, supone una alta sobrecarga de memoria para almacenar los secretos para múltiples rondas de protocolo. Además, se desperdicia ancho de banda y energía debido a la ejecución continuada de rondas del protocolo incluso cuando ningún participante tiene datos que transmitir. Otro problema

	Limitaciones	**Modelo de Atacante**		
	CPU, RAM, Otros	Global	Local	Interno
Single-proxy	↓↓	×	×	×
Mix-nets	↑↑↑	×	×	✓
Onion routing	↑↑	×	×	✓
Crowds	↓	×	×	≈
GAP	↑↑↑	✓	✓	✓
DC-nets	↑↑↑	✓	✓	✓

Table A.1: Adecuación de soluciones tradicionales

importante es que este esquema no admite múltiples emisores simultáneos, lo que restringe enormemente la aplicabilidad y la naturaleza de la red de sensores.

A.2.3 Resultados

En general, observamos que los mecanismos centralizados son menos apropiados que los distribuidos. Esto se debe a que un adversario local puede determinar los puntos de entrada al sistema de anonimato y desde allí seguir paquetes hasta el origen o el destino. Para evitar este tipo de ataques al menos sería necesario que los paquetes siguieran caminos distintos para alcanzar y salir de la red de anonimato. Si consideramos atacantes con una visión global de la red, ni siquiera con esta contramedida sería posible ocultar a los extremos de la comunicación. Los sistemas distribuidos ofrecen una mejor protección frente a este tipo de ataques ya que todos los nodos de la red formaran parte del sistema de anonimato. No obstante, no todas estas soluciones descentralizadas proporcionan un nivel de protección adecuado.

El resultado del análisis realizado se resume en la Tabla A.1, de donde se puede concluir que si bien es cierto que existen soluciones que son extremadamente costosas para las capacidades de los nodos sensores actuales, hay también otras soluciones que imponen unos requisitos o limitaciones razonables, pero que no se ajustan a las necesidades o a los modelos de atacante propios de este tipo de redes.

A.3 Estudio del estado del arte

En la sección anterior hemos concluido que las sistemas de anonimato tradicionales no son aplicables a las redes de sensores. En esta sección vamos a estudiar las características, ventajas e inconvenientes de las soluciones desarrolladas específicamente para hacer frente al problema de la privacidad de localización en redes de sensores. El objetivo final de este estudio es detectar puntos de mejora donde realizar aportaciones.

Con el fin de no extendernos demasiado en la exposición, en lugar de presentar un análisis pormenorizado de cada una de las soluciones que estudiamos en el capítulo original, en este resumen vamos a dedicarnos a destacar las características principales de los grupos de soluciones más relevantes que se establecen en nuestra taxonomía, que mostramos en la Figura A.2.

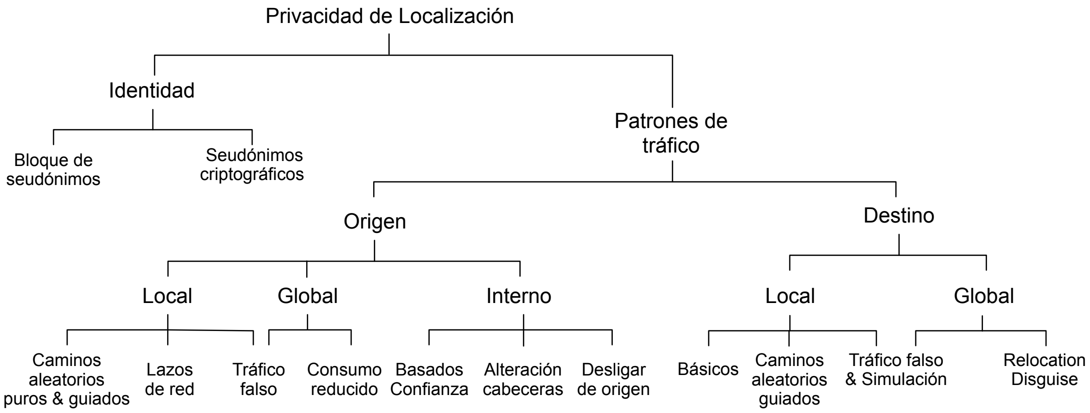


Figure A.2: Clasificación de soluciones de privacidad de localización en WSN

A.3.1 Protección de la identidad

La protección de la privacidad de localización pasa por ocultar cualquier dato sensible que se envíe a través de la red, ya sea en la carga útil de los paquetes o en sus cabeceras. Los mecanismos de confidencialidad permiten proteger los datos, sin embargo, las cabeceras de los paquetes deben ir en claro para permitir el encaminamiento de los paquetes. Por tanto, un atacante podría aprovechar esta información para determinar qué nodos intervienen en la comunicación y, con un mapa de la red, determinar su ubicación. Para dificultar esta tarea se han desarrollado esquemas de seudónimos que van cambiando la identidad del nodo de manera dinámica.

El primer grupo de soluciones se basa en un bloque de seudónimos limitado que se distribuye entre los nodos de la red de manera que la estación base tiene constancia de cuales utiliza cada nodo. El principal inconveniente de este tipo de esquemas es que se hace necesario almacenar en memoria un gran número de seudónimos. Para acabar con esta limitación, se opta por la generación de seudónimos mediante mecanismos criptográficos que permiten crear nuevos seudónimos a medida que van siendo necesarios. Normalmente la generación se lleva a cabo mediante funciones hash no invertibles para evitar que un atacante puede obtener identidades pasadas si compromete el secreto. Para evitar este tipo de problemas, se busca una solución de compromiso basada en cadenas de hashes usadas en orden inverso, lo que implica almacenar los identificadores temporalmente pero proporcionan mayor resistencia a posibles ataques. Para mayor seguridad, algunos esquemas proponen aplicar también la función hash a los secretos compartidos cada vez que se utilizan. Sin embargo, es prácticamente imposible diseñar un sistema resistente a atacantes capaces de capturar un nodo y acceder a su memoria interna, ya que tendrían acceso a todos sus secretos y comportarse como el propio nodo.

A.3.2 Protección del origen

Los mecanismos dedicados a proteger la localización de los nodos fuente de eventos dependen en gran medida del modelo de atacante considerado. Así pues, si el atacante tiene un rango de escucha local, la mayoría de soluciones se han basado en la generación de caminos aleatorios que pretenden desviar al atacante de su objetivo. Mientras que si el atacante tiene una visión global, se hace uso de tráfico falso para ocultar la presencia de eventos. Los mecanismos para hacer frente a atacantes internos o nodos comprometidos son escasos y diversos.

Atacantes locales

La estrategia seguida por atacantes locales es observar el ángulo de llegada de los mensajes para seguir el camino en sentido inverso hasta el origen. Esta estrategia tiene éxito porque, por normal general, los paquetes siguen siempre el mismo camino desde el origen hasta el destino. Por ello, la mayoría de soluciones desarrolladas hasta la fecha se basan en generar un camino diferente por cada paquete enviado, aunque también hay algunos esquemas que utilizan tráfico falso para desviar al atacante del camino de datos el mayor tiempo posible.

Básicamente encontramos dos grupos de soluciones que generan caminos aleatorios, ya sean puros o dirigidos. El primer grupo de soluciones surge a raíz del trabajo iniciado por Ozturk et al. [100], en el que se define el protocolo Phantom

Routing. Éste consta de dos fases, una primera en la que se envía el paquete de forma totalmente aleatoria y se le deja avanzar h saltos. Tras esta primera fase, se alcanza un nodo aleatorio, conocido como origen fantasma, que se encarga de enviar el mensaje a la estación base usando un algoritmo de camino óptimo. La principal limitación de este tipo de soluciones se encuentra en la primera fase y se debe a que el uso de caminos totalmente aleatorios tienden a quedarse en las proximidades del emisor. Por tanto, la longitud del camino, h, no es tan importante como su expansión. Wei-Ping et al. [144] observó, además, que es necesario evitar que los nodos fantasma se encuentren próximos entre sí o a la linea recta que pasa por el origen y la estación base, ya que esto da lugar a caminos muy similares en la segunda fase, facilitando el ataque de traceback.

Las soluciones de caminos aleatorios dirigidos surgen con el objetivo de guiar la primera fase del protocolo y acabar así con alguna de las limitaciones anteriores. Sin embargo, esto no es trivial ya que cada nodo individual suele conocer sólo a los nodos de su entorno. Dependiendo del tipo de información a la que tengan acceso los nodos, realizarán esta fase de una forma u otra. Existen soluciones, como [60, 152], donde los nodos aprovechan que saben la distancia de sus vecinos a la estación base para guiar el camino. A cada salto se elige con mayor probabilidad vecinos que estén más próximos a la estación base. De esta forma, se consiguen caminos aleatorios que acaban expandiéndose lejos del emisor. Sin embargo, esto no evita el problema que observó Wei-Ping et al.. En cambio, si los nodos tienen información sobre su ubicación y la de sus vecinos, se pueden priorizar la elección de nodos con un mayor ángulo de inclinación [140, 144], solventando así el problema. El principal problema del segundo grupo de soluciones es que, para conocer su ubicación, los nodos necesitan hardware adicional o ser colocados manualmente.

Finalmente, existen soluciones cuyo mecanismo de protección se basa en la generación de tráfico falso para desviar al atacante del camino. Entre estas, encontramos un grupo de soluciones que generan tráfico falso en forma de lazo, de manera que el tráfico real y el falso se mezclan en el lazo para confundir al atacante. En [98], los nodos que generan lazos se deciden de manera probabilística y, por tanto, los emisores no tienen porque pertenecer a un lazo, lo que hace posible llegar al emisor tras descartar lazos. Mientras que en [63], todos los nodos de la red pertenecen a lazos, con lo que los emisores pertenecen siempre a alguno de ellos. El principal problema de los lazos es que suponen un consumo energético muy elevado. Otros esquemas, como [56, 100], generan emisores falsos que envían tráfico para atraer al atacante y así desviarlos del autentico emisor. La eficacia de estas soluciones depende enormemente de conseguir una distribución balanceada de emisores falsos en la red.

Atacantes globales

Los atacantes globales tienen un rango de escucha que abarca toda la red de sensores. Esto implica que las soluciones basadas en la diversificación de caminos resulten ineficaces ante este tipo de atacantes. Para engañar a un atacante global es fundamental hacer que la transmisión de mensajes sea independiente de la detección de eventos. Por tanto, la inyección de tráfico falso es un mecanismo de defensa efectivo porque así los nodos no transmiten sólo cuando detectan un evento.

La primera propuesta para hacer frente a este tipo de atacantes fue realizada por Mehta et al. [89]. Los autores proponen que cada sensor transmita tráfico siguiendo un intervalo de tiempos fijo, de manera que, cuando se detecta un evento, el mensaje correspondiente se retrasa hasta el siguiente tiempo de envío y si no hay nada que transmitir, se manda un mensaje falso. De esta forma se consigue la no-observabilidad de los emisores. Sin embargo, esta solución impone un interesante compromiso entre el retraso en el envío de mensajes de evento y el gasto energético de realizar demasiadas transmisiones. Este desafío ha generado un importante cuerpo de investigación que ha dado lugar a soluciones que afrontan el problema desde diferentes ángulos.

Algunas soluciones se decidan a eliminar mensajes falsos con el objetivo de reducir el consumo energético [149]. Esto lo hacen mediante una serie de proxies distribuidos por la red, que al recibir mensajes comprueban si son reales o falsos. Si el mensaje es falso simplemente se desechan pero, si es real, lo almacenan temporalmente y cambian su apariencia. Si el proxy no tiene mensajes reales tendrá que generar mensajes falsos para evitar cambiar su patrón de envío. De esta forma se descarta gran cantidad de tráfico falso, pero la localización de los nodos proxy queda expuesta. Otros trabajos han tratado de reducir el consumo energético simulando la presencia de eventos en la red [89, 97]. En este caso, no basta con definir un conjunto estático de emisores falsos ya que el atacante global podría detectarlos con facilidad. Se hace necesario que los emisores falsos se vayan desplazando por la red de manera que su patrón de movimiento se asemeje lo más posible al tipo de eventos que se desea ocultar. Es precisamente esta, la mayor limitación de estas soluciones, que requieren un conocimiento muy profundo del comportamiento de los eventos monitorizados.

Finalmente, una serie de trabajos se ha centrado en reducir la latencia en el envío de los eventos al tiempo que se mantiene un nivel de protección casi perfecto [129]. La idea es que, en una ventana de tiempos inter-mensaje con una determinada distribución de probabilidad, es posible adelantar el envío de un mensaje de evento siempre y cuando los parámetros de la distribución no se vean alterados. A tal fin, cuando un nodo detecta un evento calcula cuál es el mínimo tiempo en el que puede enviar, de manera que la ventana de tiempos pasa un test de bondad estadística. A la larga, esto puede alterar la media de tiempos de la distribución por lo que es necesario retrasar el envío de mensajes falsos posteriores. Sin embargo, gracias a este mecanismo de recuperación de la media un atacante podría detectar diferencias entre diferentes ventanas de tiempo. Por lo que, en [8], se propone introducir cierta dependencia estadística también entre mensajes falsos haciendo que se parezcan a ventanas con mensajes reales.

Atacantes internos

Los atacantes internos son aquellos nodos legítimos de la red que han sido comprometidos por el atacante y trabajan a sus órdenes. La ventaja principal de este tipo de atacantes es que, al ser parte del sistema comparten secretos y tienen acceso al contenido de los paquetes que atraviesan su entorno.

Las soluciones propuestas para hacer frente a este tipo de atacante son, hasta la fecha, muy limitadas y de naturaleza diversa. En primer lugar, encontramos una solución que utiliza la noción de confianza para evitar que los paquetes pasen por nodos comprometidos [124]. Cada nodo agrupa sus vecinos en varios grupos según su distancia a la estación base. Cuando tiene que enviar un paquete, elige aleatoriamente entre los nodos más cercanos que le resultan confiables. Si no encuentra ninguno, busca en el resto de grupos por si hubiera alguno confiable. Asimismo, proponen el remplazo de los identificadores de la cabecera antes de reenviar el paquete. En [104] se propone un esquema de transformación de las cabeceras más sofisticado, que consiste en la aplicación de determinadas operaciones criptográficas en nodos de la ruta seleccionados dinámicamente.

Finalmente, existe una solución que trata de evitar que un atacante pueda obtener información de localización en el momento que compromete un nodo [130].

Esto tiene sentido en un determinado tipo de redes de sensores en las que la estación base aparece esporádicamente para recuperar los datos. En este tipo de redes, hay determinados sensores encargados de almacenar temporalmente los datos recogidos. La idea de la solución es que cuando un nodo detecta un evento, cifra los datos con una clave compartida con la estación base y envía el resultado a uno de estos nodos de almacenamiento. De esta forma, si un atacante compromete un nodo y recupera su contenido, no será capaz de descifrar los datos puesto que las claves de cifrado no se encuentran en el nodo.

A.3.3 Protección del Destino

Al igual que para la protección de los nodos origen, los mecanismos para proteger la ubicación de la estación base dependen del modelo atacante. En general, la idea es homogeneizar la tasa de transmisión de los nodos de la red con el fin de evitar que la zona próxima a la estación base tenga una tasa de transmisiones muy superior al resto de la red.

Atacantes locales

Los atacantes locales utilizan dos tipos de estrategias que les permiten determinar la dirección hacia la estación base: la correlación de tiempos y la monitorización de la tasa de envío. La primera se basa en que los nodos reenvían el tráfico que reciben inmediatamente después recibirlo. La segunda tiene como fundamento que los nodos próximos a la estación base tienen una mayor tasa de envío.

Existen una serie de medidas básicas que permiten aliviar estos problemas [39]. En primer lugar, cambiar la apariencia de los mensajes a cada salto, para lo que se descifran los paquetes recibidos y posteriormente se vuelven a cifrar con la clave del siguiente salto. A pesar de esto, un atacante puede determinar la dirección del flujo de datos observando los tiempos de envío entre nodos vecinos. Para evitar este tipo de ataques de correlación de tiempos, se puede optar por almacenar temporalmente los paquetes recibidos o añadir pequeños retrasos. Sin embargo, el atacante puede conocer el siguiente salto observando qué nodos reciben un mayor número de paquetes. Estas medidas son insuficientes y se hace necesario el empleo de técnicas más sofisticadas para balancear la carga de tráfico en la red.

En primer lugar, nos encontramos con soluciones que utilizan caminos aleatorios guiados hacia la estación base. En [38], los mensajes se envía siempre hacia vecinos más próximos a la estación base. El problema es que, tras varias observaciones, el atacante es capaz de determinar qué nodos están más próximos. Una alternativa es enviar los paquetes utilizando caminos uniformemente aleatorios pero la latencia de las comunicaciones se hace insoportable. Por ello, la mayoría de soluciones [38, 58] optan por introducir tráfico falso en otras direcciones para que la tendencia hacia la estación base no sean tan clara. El problema es que la cantidad de tráfico falso o la forma de inyectarlo hace que el atacante siga siendo capaz de determinar la dirección hacia su objetivo.

La inyección de tráfico falso también se ha utilizado para simular la presencia de la estación base en diversas zonas de la red [25, 38]. En lugar de enviar el tráfico de forma aleatoria, se dirige sólo a determinadas zonas, con lo que se consigue que éstas tengan una elevada concentración de tráfico. El principal inconveniente de este tipo de soluciones es que requiere enviar tanto tráfico falso a cada una de esas zonas como recibe la estación base, para así atraer los atacantes. Esto supone un coste energético exageradamente elevado. Por otra parte, para que este tipo de técnicas tenga éxito es necesario que las zonas que reciben tráfico falso estén uniformemente distribuidas en la red.

Atacantes globales

Los atacantes globales son capaces de localizar la estación base porque la concentración de mensajes en su entorno es mayor que en cualquier otra zona de la red. Por ello, el mecanismos fundamental para evitar este tipo de atacante es generar tráfico falso aunque también se han desarrollado otro tipo de técnicas relevantes.

En primer lugar, es interesante observar que un protocolo de inundación tradicional proporciona el mejor nivel de protección, ya que cada vez que un nodo transmite un mensaje, todos los nodos de la red lo retransmiten, consiguiendo así una tasa de envío uniforme en toda la red. Sin embargo este modo de funcionamiento también supone un gasto energético inmanejable. Por ello, algunas soluciones han tratado de reducir el coste de esta solución reduciendo el área de inundación [90]. Así, si todos los paquetes se envían a esta zona y obligamos a la estación base a estar en sus inmediaciones, se consigue que la estación base reciba los paquetes sin necesidad de inundar toda la red. Obviamente, al reducir el área de inundación también disminuye la protección de la estación base.

La mayoría de soluciones restantes se basan en el uso de tráfico falso. Por ejemplo, hay soluciones que tratan de equilibrar la tasa de transferencia de todos los nodos de la red independientemente de su distancia a la estación base [155, 156]. Esto se consigue inyectando tráfico falso en función de la tasa de transferencia de los vecinos directos de la estación base. Como estos nodos son los que más mensajes envían, el resto de nodos trata de compensar su tasa de envío para transmitir tanto como estos. La principal limitación de estos trabajos es que se asume que todos los nodos tienen una tasa de envío constante lo que permite estimar cuantos mensajes mandarían los vecinos de la estación base. Además, cabe la duda de si este mecanismo merece la pena frente a un protocolo de inundación tradicional.

También cabe destacar las técnicas que ocultan la estación base por similitud. La idea es que la estación base se comporte como un nodo cualquiera de la red, de manera que cuando reciba un paquete lo reenvíe para que viaje varios saltos alejado de su posición [1]. El problema de esta solución es que aunque los paquetes no muera en la estación base, igualmente la atravesarán, con lo que esta zona seguirá teniendo el mayor volumen de tráfico. Otra técnica interesante consiste en reubicar a la estación base en otra posición considerada más segura [1, 39]. El principal inconveniente de este tipo de soluciones es cómo tener la certeza de que una posición es realmente segura. Si la estación base calcula esta posición en función de la tasa de envío en diferentes zonas, el atacante también podría realizar un cálculo similar y estimar la siguiente posición.

Finalmente, la creación de zonas calientes con un alto volumen de tráfico también ha sido utilizada como mecanismo de defensa ante atacantes globales [24, 90]. En este caso, hay que prestar especial cuidado a la creación de nuevas zonas que reciben tráfico falso ya que, en cualquier momento, la zona donde se encuentra la estación base será una zona caliente. Así pues, el atacante podría fácilmente realizar un ataque de intersección entre los conjuntos de zonas observadas en diferentes periodos de tiempo e ir reduciendo el número de zonas calientes falsas.

A.4 Mecanismo CALP para la protección del origen

Nuestra aportación a la protección de la localización de nodos fuente es el mecanismo CALP (*Context-Aware Source-Location Privacy*). Tras el estudio del estado del arte observamos que uno de los principales problemas que adolece a la mayoría de soluciones es que su consumo energético es elevado y esto se debe en gran parte a que el mecanismo de protección está activo durante todo el tiempo de vida de la red. El objetivo que nos marcamos con CALP fue si era posible activar el mecanismo de defensa sólo cuando la privacidad de localización se viera amenazada, es decir, cuando el atacante se encuentra en el área de despliegue de la red.

CALP aprovecha la capacidad sensorial de la red para detectar posibles atacantes y hacer que los mensajes se desvíen de su trayectoria para evitar el área donde se encuentran estos. De esta forma se limita el número de paquetes que son capaces de capturar y, por tanto, se reduce su probabilidad de llegar al objetivo. Al mismo tiempo, se reduce considerablemente la sobrecarga impuesta por las soluciones actuales, que repercuten negativamente en el tiempo de entrega y el consumo energético.

A.4.1 Escenario y atacante

Consideramos redes de sensores tradicionales, utilizadas para la monitorización y que siguen un modelo de transmisión basado en eventos. La red está compuesta por un número importante de nodos que se encuentran uniformemente distribuidos en una amplia extensión de terreno. Además, los nodos son conscientes de todos sus vecinos y comparten secretos con ellos para permitir un cifrado seguro de las comunicaciones. La suposición más importante para el funcionamiento de nuestro esquema es que cada nodo es capaz de detectar la presencia de objetos en su entorno. Esto se puede conseguir por varios medios, ya sean sensores de infrarrojos, acústicos, de presión, magnéticos, etc.

Por otra parte, consideramos que el modelo de atacante es un observador local, con un rango de escucha similar al de un nodo sensor tradicional, y que se desplaza en el terreno realizando ataques de seguimiento de paquetes en sentido inverso. Permitimos al atacante comenzar desde la posición más favorable para él, que es junto a la estación base. Consideramos que puede seguir dos tipos de estrategias: ser paciente o curioso. El adversario paciente espera hasta que escucha un paquete, en cuyo caso se mueve hacia el emisor del mismo. En caso de estar mucho tiempo sin detectar un paquete vuelve a su posición anterior para ver si desde esa posición vuelve a escuchar algo. Eventualmente, el atacante podría volver a su posición original, junto a la estación base. El atacante curioso se comporta de manera parecida, pero en lugar de esperar a que lleguen paquetes se empieza a mover de manera aleatoria hasta que detecta alguno.

A.4.2 Descripción de la solución

El mecanismo CALP puede verse como un complemento software que se integra con el resto de componentes de un nodo sensor para permitir protocolos de encaminamiento respetuosos con la privacidad. Una de las ventajas principales de utilizar CALP es que no requiere grandes modificaciones o reemplazar el protocolo de encaminamiento utilizado en la red, ya que la interacción entre ambos se realiza de manera prácticamente transparente a través de las tablas de rutas de los nodos.

En líneas generales, el funcionamiento de CALP es como sigue. Ante la detección de un atacante en su entorno, los nodos reaccionan enviando un mensaje de actualización de rutas a sus vecinos. Este mensaje es reenviado durante varios saltos desde la posición donde se detectó al atacante y se utiliza para modificar las tablas de rutas de los nodos, de manera que cuando se tenga que enviar un mensaje de evento se evite la región controlada por el atacante.

Para la detección del atacante, la red puede basarse en la información ofrecida por sus propios sensores o puede hacer uso de técnicas más complejas basadas en la interferencias que generan objetos en movimiento sobre las señales de radio [146]. Esto permite a la red determinar la presencia de objetos en su entorno pero no es posible discriminar si se trata de un atacante u otra entidad. Cuando la red se encarga, por ejemplo, de monitorizar tropas en un campo de batalla, éstas llevarán consigo algún dispositivo que los identifique mediante un protocolo de desafío-respuesta o similar [136]. En tal caso, y para evitar problemas, se puede optar por activar el mecanismo de defensa ante cualquier otra presencia en el terreno, aunque en ocasiones se trate de un falso positivo. Para reducir la tasa de falsos positivos, la red puede observar el comportamiento de los objetos en presencia de mensajes. Si una entidad no reconocida sigue paquetes en sentido contrario, la probabilidad de que se trate de un atacante es elevada y por tanto se comienza a enviar mensajes de actualización de rutas.

Dado que el objetivo es anticiparse a los movimientos del atacante, es necesario que los mensajes de actualización de rutas se expandan varios saltos. De esta forma, tanto los vecinos inmediatos como los cercanos tendrán constancia de la distancia a la que se encuentra el adversario. Así pues, cuando un nodo detecta una presencia, crea un paquete de actualización indicando a sus vecinos que el atacante se encuentra a distancia 1 de ellos. Estos nodos almacenan el valor recibido, incrementan el valor de distancia y repiten el proceso, difundiendo de este modo la alerta. Cuanto mayor sea el rango de escucha del atacante más tendrá que extenderse la alerta para permitir que los paquetes sean enviados fuera de su alcance. Es importante notar que el envío de este tipo de paquetes no aporta ningún tipo de información al atacante acerca de la localización de nodos fuente ya que su transmisión es independiente de la presencia de eventos en la red. Y para evitar un consumo extra en la red, se puede optar por aprovechar el envío periódico de tramas baliza (i.e., beacon frames), cuyo cometido es informar de parámetros de configuración de la red.

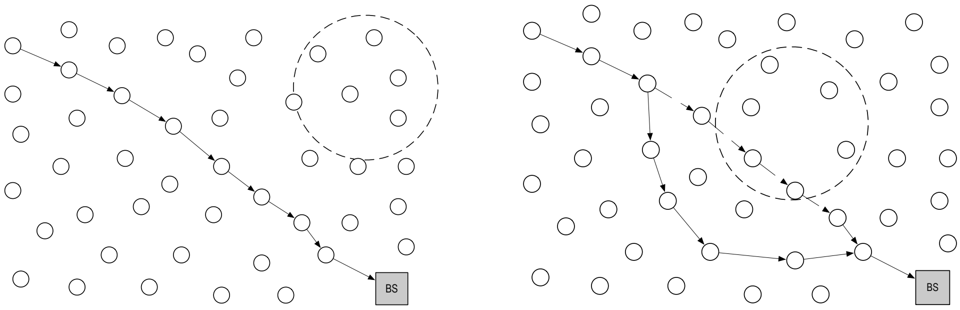
Por último, durante la fase de envío de información, cada nodo decide el siguiente salto en función del protocolo de encaminamiento utilizado y la distancia de sus vecinos al atacante. Dependiendo de cómo se utilice esta información de distancia al atacante, tenemos al menos dos formas de enviar paquetes: impidiendo que los nodos envíen a vecinos que se encuentren a menos de una *distancia mínima de seguridad* o penalizando la selección de nodos dentro de esta región. Así pues, tendremos una distancia de seguridad estricta y otra permisiva.

A.4.3 CALP con camino óptimo

El mecanismo CALP puede utilizarse con diferentes protocolos de encaminamiento para mejorar la protección de la privacidad en WSNs. En esta sección nos centramos en la aplicación de nuestra solución a protocolos de camino óptimo (i.e., *shortest-path routing*) ya que proporcionan características interesantes como latencia y consumo energético mínimos.

El algoritmo de base elegido para este trabajo toma decisiones de encaminamiento de forma voraz, ya que selecciona vecinos de manera localmente optima. Un vecino es localmente óptimo si es el que menos se desvía del segmento imaginario que conecta al emisor con la estación base. De esta forma, al combinarlo con CALP se consigue que los paquetes sigan el camino más corto, desviándose lo menos posible del camino óptimo, incluso en presencia del atacante. En la Figure A.3 se muestra el proceso de adaptación del camino ante la presencia de un atacante en el área que atraviesa el camino más corto a la estación base.

Se han desarrollado dos versiones del protocolo. En la versión estricta, se bloquea el envío a nodos que se encuentran próximos al atacante. Para ello,



(a) (b)

Figure A.3: Adaptación del camino en presencia del atacante

cuando un nodo tiene datos que enviar, obtiene los vecinos de la tabla de rutas y para cada uno de ellos observa si su distancia respecto al atacante es adecuada. Si es menor que la distancia mínima de seguridad, aplica la máxima penalización para evitar que sea elegido, en caso contrario aplica una penalización lineal en función de su ángulo y su distancia. Finalmente, el nodo elegido es el que tenga la menor penalización. Este comportamiento se describe en el Algoritmo 5

Algorithm 5 Envío en CALP estricto

```
Entrada: $DIST\_MIN\_SEG$
Entrada: $datos$
$vecinos \leftarrow obtener\_vecinos()$
for all $n_i \in vecinos$ do
    if $distancia(n_i) \leq DIST\_MIN\_SEG$ then
        $penalización[n_i] = \infty$
    else
        $penalización[n_i] = ángulo(n_i) + \pi/distancia(n_i)$
    end if
end for
$destino \leftarrow minimo(penalización, vecinos)$
$enviar(datos, destino)$
```

En la versión permisiva, los paquetes no son bloqueados pero los vecinos que se encuentran en el interior del área considerada peligrosa reciben una penalización mayor. De esta forma, los paquetes sólo son desviados de su camino si la penalización por desviar es menor que por entrar en el área controlada por el atacante. Una descripción detallada del funcionamiento puede verse en el Algoritmo 6. Tras obtener su lista de vecinos, el nodo calcula la penalización base en función de los ángulos y distancia al atacante de sus vecinos. Esta penalización es incrementada en un factor que es inversamente proporcional a la distancia al atacante si éste se encuentra a menos distancia que la distancia mínima de seguridad. El vecino con menos penalización recibe los datos.

Algorithm 6 Envío en CALP permisivo

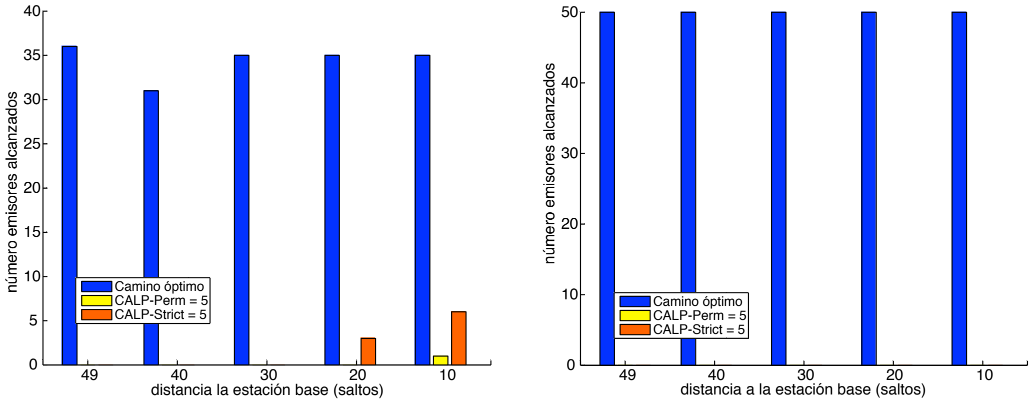
```
**Entrada:** $DIST\_MIN\_SEG$
**Entrada:** $datos$
$vecinos \leftarrow obtener\_vecinos()$
**for all** $n_i \in vecinos$ **do**
    $penalización[n_i] = ángulo(n_i) + \pi/distancia(n_i)$
    **if** $distancia(n_i) \leq DIST\_MIN\_SEG$ **then**
        $penalización[n_i] = penalización[n_i] + 1/distancia(n_i)$
    **end if**
**end for**
$destino \leftarrow minimo(penalización, vecinos)$
$enviar(datos, destino)$
```

Para ambos algoritmos ocurre que, cuando el atacante no está presente en el terreno o en la zona próxima al camino, el algoritmo se comporta como el protocolo original de camino óptimo. Finalmente, es interesante observar que las operaciones realizadas son extremadamente livianas, incluso para dispositivos muy restringidos. Asimismo, la memoria requerida por este esquema es insignificante. Básicamente, necesita sólo de una nueva columna en la tabla de rutas para mantener la distancia de cada vecino al atacante.

A.4.4 Evaluación

La evaluación de nuestra solución se ha llevado a cabo en nuestro propio simulador desarrollado en Matlab. Cada instancia de simulación cuenta con una red de sensores de 100×100 sensores, en la que por cada uno de los 500 eventos programados en el simulador se envía un nuevo paquete. En cada evento, el atacante es capaz de desplazarse un salto. Consideramos tanto atacantes pacientes como curiosos, que originalmente se encuentran junto a la estación base. Por defecto la distancia mínima de seguridad está fijada a una distancia de 5 saltos.

El nivel de protección de la solución se mide en función del número de nodos origen que el atacante es capaz de alcanzar tras 50 ejecuciones de la misma instancia de simulación y se compara con el resultado de utilizar el algoritmo de encaminamiento por camino óptimo. Los resultados obtenidos se muestran en la



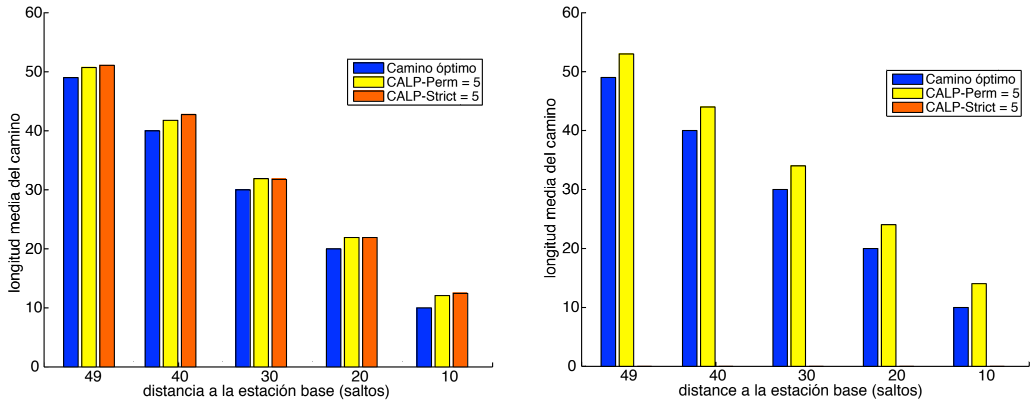
(a) Atacante curioso

(b) Atacante paciente

Figure A.4: Número de emisores alcanzados

Figura A.4. A partir de esta figura, observamos que el algoritmo de camino óptimo permite al atacante alcanzar la estación base siempre que sea paciente y espere a recibir el primer paquete, puesto que el resto seguirá la misma ruta. Cuando se utiliza un algoritmo de camino óptimo en combinación con el mecanismo CALP, la situación mejora considerablemente. Ante un adversario curioso (Figura A.4a), el atacante tiene una ligera ventaja si los emisores se encuentran próximos a la estación base y sorprendentemente la versión permisiva ofrece mejor protección que la estricta. Esto se debe a que, en la versión estricta, el atacante se mueve libremente ya que no observa paquetes en su entorno. En cambio, con la versión permisiva, el atacante sigue algunos paquetes que acaban por confundirlo. Ante un atacante paciente (Figura A.4b), no se ofrece información de localización. En la versión permisiva, los paquetes atraen al atacante a una zona lejos de la estación base y después los caminos se readaptan para evitar al atacante y enviar los datos a su destino. En cambio, con la versión estricta, al bloquearse el envío de paquetes en el entorno del atacante, éste no se ve atraído por los paquetes y por tanto no abandona su posición original.

Para evaluar el funcionamiento de nuestro protocolo nos fijamos en la longitud de los caminos generados y, de nuevo, lo comparamos con el protocolo de camino óptimo, que supone una cota máxima de eficiencia. En general, observamos que la longitud promedio del camino (Figura A.5) es ligeramente superior a la del camino óptimo, siendo esta diferencia mayor cuando el atacante es paciente. En ese caso, cuando la estrategia es estricta observamos que no se representa la longitud promedio puesto que la estación base no llega a recibir paquetes. Una



(a) Atacante curioso

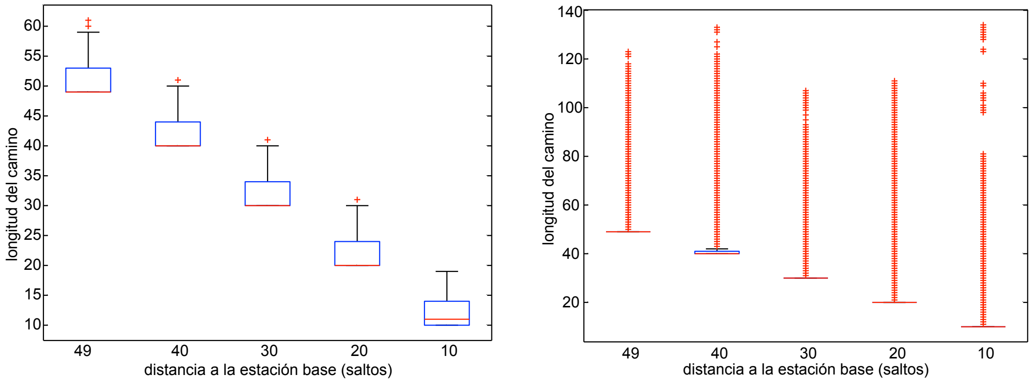
(b) Atacante paciente

Figure A.5: Longitud media del camino

solución a este problema pasa por desarrollar una versión mixta del protocolo, que haga un uso permisivo de la distancia de seguridad cuando los nodos están próximos a la estación base, y estricto cuando el paquete atraviesa zonas remotas.

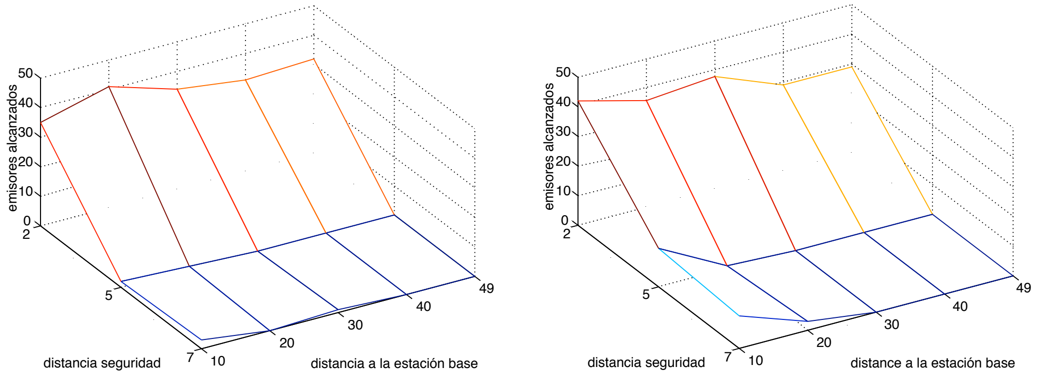
A pesar de obtener unos valores promedio bastante razonables para ambas versiones del protocolo, sería interesante saber si hay ocasiones en las que estos caminos pueden retrasarse demasiado con el fin de ver la idoneidad de nuestra solución a aplicaciones con requisitos de tiempo real. Para ello, hemos estudiado la longitud de los caminos, en presencia de un atacante curioso, mediante diagramas de cajas. Observamos que, cuando se utiliza la versión permisiva de CALP (Figura A.6a), la mayoría de paquetes viaja una número similar de saltos y sólo en raras ocasiones hay algún paquete cuya entrega se demora. Sin embargo, cuando se utiliza la versión estricta del protocolo (Figura A.6a), hay paquetes que llegan a triplicar la distancia media de la distribución. Esto se debe irremediablemente a la presencia del atacante en zonas próximas a la estación base, lo que provoca que los paquetes no puedan ser entregados. Para evitar el coste energético que esto supone, los nodos podrían mantener una lista de nodos visitados y en caso de generar un bucle, almacenar temporalmente el paquete. Asimismo, prevemos que la opción de utilizar una estrategia mixta podría aliviar igualmente el problema.

Por último, hemos estudiado cómo afecta la distancia mínima de seguridad al nivel de protección y a la longitud de los caminos. Para ello, hemos considerado un atacante curioso y hemos definido tres distancias de seguridad, con valores 2, 5 y 7. Como era de esperar, cuanto mayor es la distancia de seguridad, menor número de capturas y, por tanto, mejor es la protección (Figura A.7). Además,



(a) CALP permisivo (b) CALP estricto

Figure A.6: Distribución de longitudes de caminos

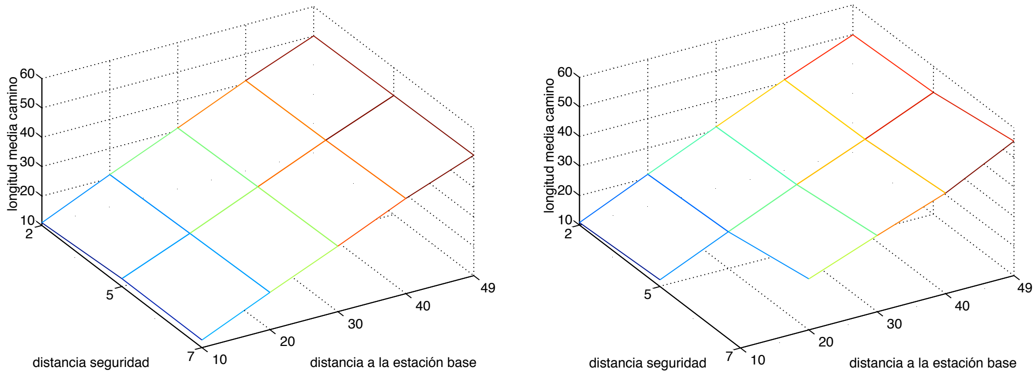


(a) CALP permisivo (b) CALP estricto

Figure A.7: Impacto sobre el número de emisores alcanzados

observamos que, en ambos casos, un perímetro de seguridad mayor a 2 proporciona un nivel adecuado de protección, siendo más eficaz cuando la versión del protocolo es permisiva (Figura A.7a). Por otra parte, como puede verse en la Figura A.8, la distancia de seguridad tiene un impacto casi despreciable sobre la longitud promedio de los caminos. Sin embargo, como vimos anteriormente, pueden existir paquetes que atraviesen un número elevado de saltos antes de ser entregados. Además, se aprecia que la versión estricta es más sensible a la distancia de seguridad (Figura A.8b). Por último, es interesante observar que cuando la distancia de seguridad es 7 y el nodo se encuentra a 10 saltos del destino, no se entregan los paquetes.

En general, podemos concluir que la versión estricta del protocolo tiene la ventaja de asegurar que el atacante no es capaz de capturar paquetes pero, por



(a) CALP permisivo

(b) CALP estricto

Figure A.8: Impacto sobre la longitud media de caminos

contra, en ocasiones implica un retraso excesivo en la entrega de los paquetes. Incluso puede que la entrega no llegue a producirse cuando el atacante no abandona las proximidades de la estación base. La versión permisiva tiene la ventaja de que los paquetes siempre se entregan en un tiempo muy próximo al óptimo y, por tanto, es adecuada para aplicaciones con requisitos de tiempo real. Sin embargo, tiene el inconveniente de que el atacante puede capturar paquetes en su entorno aunque esto no llega a tener serias consecuencias en el nivel de protección proporcionado a los emisores. Una estrategia mixta que combine las bondades de ambas estrategias puede resultar bastante interesante.

A.5 Mecanismo HISP-NC para la protección del destino

En esta sección presentamos HISP-NC (*Homogeneous Injection for Sink Privacy with Node Compromise protection*), nuestro mecanismo para proteger a la estación base de posibles atacantes locales. Nuestro objetivo era mejorar el nivel protección proporcionado por las soluciones basadas en caminos aleatorios, que en ocasiones revelan la dirección hacia la estación base. Asimismo, observamos que ninguna de las soluciones existente ofrece protección frente a atacantes capaces de obtener las tablas de rutas, a pesar de que éstas contienen información que permiten encontrar la estación base.

El protocolo HISP-NC consta de dos esquemas complementarios. El primer esquema, es utilizado durante la transmisión de datos y se basa en el envío de mensajes siguiendo un camino aleatorio hacia la estación base. Para ocultar el flujo de datos, se utilizan cantidades controladas de tráfico falso que permiten homogeneizar el número de paquetes enviado a los vecinos de cada nodo. El segundo esquema, se utiliza para reducir el riesgo que entraña la captura de las tablas de rutas de los nodos. Se trata de un esquema de perturbación que modifica las tablas de rutas al tiempo que se asegura que los paquetes son entregados a la estación base. Aunque son esquemas complementarios, si el riesgo de recibir ataques físicos es reducido, el primero puede funcionar de manera autónoma para reducir la sobrecarga que supone la perturbación.

A.5.1 Escenario y atacante

Consideramos una red de sensores dedicada a la monitorización de eventos, que está compuesta por un gran número de sensores y una única estación base. Asumimos que la conectividad de la red es elevada y que cada nodo conoce a sus vecinos así como su distancia a la estación base. De esta forma, cada nodo puede construir su tabla de rutas ordenada en función de la distancia: vecinos en más cercanos, a la misma distancia, o más alejados. Nos referiremos a cada uno de estos grupos como $L^{\mathcal{C}}$, $L^{\mathcal{E}}$ y $L^{\mathcal{F}}$ respectivamente. Además, suponemos que los nodos comparten secretos para establecer enlaces seguros.

El atacante es capaz de realizar tanto ataques pasivos (análisis de tráfico) como activos (captura de nodos). El rango de escucha del atacante es similar al de un nodo sensor cualquiera. Tras observar las comunicaciones en su entorno, el atacante decide moverse con el fin de reducir su distancia hasta el destino. Esta decisión depende de si el atacante opta por un ataque por correlación de tiempos (*time-correlation*) o un ataque por volumen de tráfico (*rate monitoring*). Cuando el atacante realiza ataques activos se limita a inspeccionar las tablas de rutas para conocer los vecinos que están más cercanos a la estación base. En la literatura no existe una estrategia de captura claramente definida y mientras que algunos autores consideran la captura aleatoria de nodos, otros optan por la captura de nodos en una región. En este trabajo consideramos que el atacante es más exitoso si centra su esfuerzo en una región y avanza según la información obtenida. Dado el esfuerzo y el riesgo que supone un ataque de este tipo, el atacante sólo podrá comprometer un número reducido de nodos.

A.5.2 Descripción de la solución

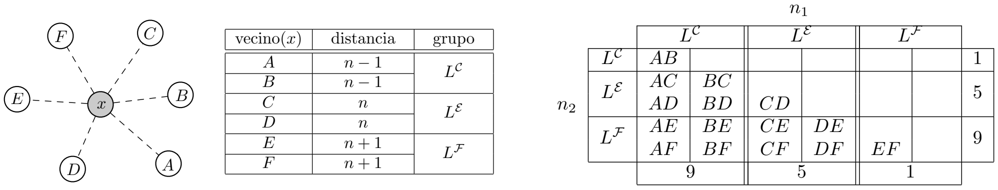
El protocolo de transmisión es básicamente un camino aleatorio guiado cuya dirección se oculta inyectando tráfico falso de manera controlada. Cuando un nodo tiene datos que enviar, éste transmite el paquete hacia la estación base con cierta probabilidad sesgada. El envío va acompañado de otro paquete falso que homogeneiza la tasa de paquetes enviados a cada vecino. De esta forma se oculta el flujo de datos real localmente sin introducir un retraso excesivo en las comunicaciones.

El algoritmo de perturbación consiste en reordenar la tabla de rutas de cada nodo para que si un atacante tiene acceso a ésta no sea capaz de alcanzar fácilmente la estación base al tener la certeza de que los nodos más próximos se encuentran más altos en la tabla. El nivel de perturbación de la tabla introduce incertidumbre en el atacante pero al mismo tiempo repercute negativamente en el tiempo de llegada de los paquetes.

Protocolo de transmisión

Cada vez que un nodo transmite, envía dos paquetes. El mecanismo ideado se basa en la elección de parejas de vecinos a partir de las combinaciones sin repetición de parejas de vecinos de la tabla de rutas del nodo. Si la tabla de rutas se encuentra ordenada en función de la distancia a la estación base (i.e., $\{L^{\mathcal{C}}, L^{\mathcal{E}}, L^{\mathcal{F}}\}$), el primer elemento de cada una de las combinaciones resultantes pertenece a $L^{\mathcal{C}}$ con alta probabilidad. Por tanto, si los paquetes reales se envían a este nodo, conseguimos asegurar la convergencia de los datos a la estación base. Si además las combinaciones se eligen de manera uniformemente aleatoria, todos los vecinos reciben en promedio el mismo número de paquetes, ya que cada vecino aparece exactamente en $l-1$ combinaciones, siendo l el número de vecinos de la tabla. Como los mensajes reales y falsos son indistinguibles, un observador local es incapaz de distinguir el flujo real del falso.

En la Figura A.9a mostramos la tabla de rutas de un nodo arbitrario (la columna grupo se añade sólo con fines aclaratorios) y, en la Figura A.9b, las combinaciones que resultan de esta tabla. Se puede observar que se obtienen 9 combinaciones donde el primer elemento de la combinación, n_1, es un vecino más cercano a la estación base, 5 donde n_1 está a la misma distancia que el nodo x, y 1 combinación donde n_1 se encuentra en la dirección opuesta a la estación base.



vecino(x)	distancia	grupo
A	$n-1$	$L^{\mathcal{C}}$
B	$n-1$	
C	n	$L^{\mathcal{E}}$
D	n	
E	$n+1$	$L^{\mathcal{F}}$
F	$n+1$	

(a) Tabla de rutas del nodo x

				n_1				
		$L^{\mathcal{C}}$		$L^{\mathcal{E}}$		$L^{\mathcal{F}}$		
	$L^{\mathcal{C}}$	AB						1
n_2	$L^{\mathcal{E}}$	AC AD	BC BD	CD				5
	$L^{\mathcal{F}}$	AE AF	BE BF	CE CF	DE DF	EF		9
		9		5		1		

(b) Combinaciones resultantes

Figure A.9: Proceso de selección de vecinos en HISP-NC

De esta forma, al elegir una combinación al azar, la probabilidad de enviar el paquete real hacia un nodo más próximo a la estación base es precisamente 9/15.

En el Algoritmo 7 se muestra el comportamiento de nuestro protocolo de transmisión. Los argumentos de entrada son el paquete a reenviar, las combinaciones sin repetición de la tabla de rutas ordenada y el parámetro TTL_FALSO, que controla el tiempo de vida de los mensajes falsos en la red y que depende del rango de escucha del adversario. Cuando un nodo recibe un paquete real, se elige una combinación aleatoria de dos vecinos que recibirán el mensaje real y uno falso (líneas 1 a 3). El mensaje falso se reenviará durante TTL_FALSO saltos. Si el paquete recibido es un paquete falso aún vigente, se reduce su tiempo de vida y se envían dos mensajes falsos (líneas 5 a 7). Además, las parejas de paquetes se envían en un orden aleatorio para evitar que el atacante puede determinar de forma trivial cuál de los paquetes es el real.

Algorithm 7 Transmisión en HISP-NC

```
Entrada: $paquete \leftarrow recibir()$
Entrada: $combs \leftarrow combinar(\{L^{\mathcal{C}}, L^{\mathcal{E}}, L^{\mathcal{F}}\}, 2)$
Entrada: $TTL\_FALSO$
1: $\{n1, n2\} \leftarrow selección\_aleatoria(combs)$
2: if $es\_real(paquete)$ then
3:     $enviar\_aleatorio(n1, paquete, n2, paquete\_falso(TTL\_FALSO))$
4: else
5:     $TTL \leftarrow obtener\_tiempo\_vida(paquete) - 1$
6:     if $TTL > 0$ then
7:         $enviar\_aleatorio(n1, paquete\_falso(TTL), n2, paquete\_falso(TTL))$
8:     end if
9: end if
```

Perturbación de tablas

Mantener el orden de las tablas de rutas es fundamental para el correcto funcionamiento de nuestro protocolo de transmisión. Sin embargo, esto puede permitir a un atacante determinar qué vecinos se encuentran más próximos a la estación base con sólo capturar el nodo y obtener su tabla de rutas. Por ello, ante situaciones de riesgo, es fundamental crear cierta incertidumbre aunque esto conlleve un aumento en el tiempo de entrega de los paquetes.

Una tabla de rutas $L^* = L^{\mathcal{C}} \cup L^{\mathcal{E}} \cup L^{\mathcal{F}}$ es una ordenación concreta de los vecinos de un nodo. Para que nuestro protocolo de transmisión funcione, la ordenación de la tabla de rutas debe cumplir ciertas propiedades y en tal caso diremos que tiene un sesgo correcto. Una tabla está correctamente sesgada si cumple que $\mathbb{P}(n_1 \in L^{\mathcal{C}}) > \mathbb{P}(n_1 \in L^{\mathcal{F}})$, es decir, la probabilidad de elegir una combinación en la que el primer elemento, n_1 está más próximo a la estación base, es mayor que la probabilidad de mandarlo a un nodo más alejado.

Podemos cuantificar el sesgo de una tabla de rutas, $bias(r) \in [-1, 1]$, en función de la posición que cada vecino tiene en la tabla. Esto es así porque el número de combinaciones en las que un vecino aparece en primera posición coincide exactamente con el número de vecinos que están debajo suya en la tabla. Por ejemplo, en la Figura A.9, el vecino A aparece en 5 combinaciones como primer elemento, mientras que el vecino F no aparece en ninguna de las combinaciones como primer elemento. El sesgo nos permite estimar la velocidad a la que avanzan los paquetes reales a la estación base. Cuanto más próximo a 1 es el sesgo, más probable es que el siguiente nodo de la ruta se encuentre más próximo a la estación base, mientras que valores próximos a -1 indican que el siguiente nodo se encontrará con gran probabilidad más alejado. Formalmente puede calcularse como:

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

donde $C = 1 + 2 + \ldots + (N-1)$ es el total de combinaciones que resultan de la tabla. Es sencillo comprobar que si $L^* \equiv L^{\mathcal{F}}$, es decir, el nodo sólo tiene vecinos más alejados, entonces $bias(r) = -1$ ya que $\sum\limits_{n \in L^{\mathcal{F}}} pos(n) = C$. Del mismo modo, si $L^* \equiv L^{\mathcal{C}}$, entonces $bias(r) = 1$.

Nuestro algoritmo de perturbación recibirá como parámetros un valor de sesgo deseado y una tabla de rutas, y devolverá la tabla reordenada conforme al sesgo dado. En el Algoritmo 8 puede observarse que hemos modelado este algoritmo como un problema de optimización, donde la función objetivo (línea 1) depende del valor de sesgo deseado y la ordenación actual de la tabla. En concreto, el algoritmo se inspira en estrategias evolutivas donde intercambiamos dos elementos de la tabla de rutas (línea 4) y comprobamos si así se reduce la distancia al sesgo deseado (línea 6). El proceso se repite por un número máximo de iteraciones o bien hasta que se genere una ordenación acorde al sesgo.

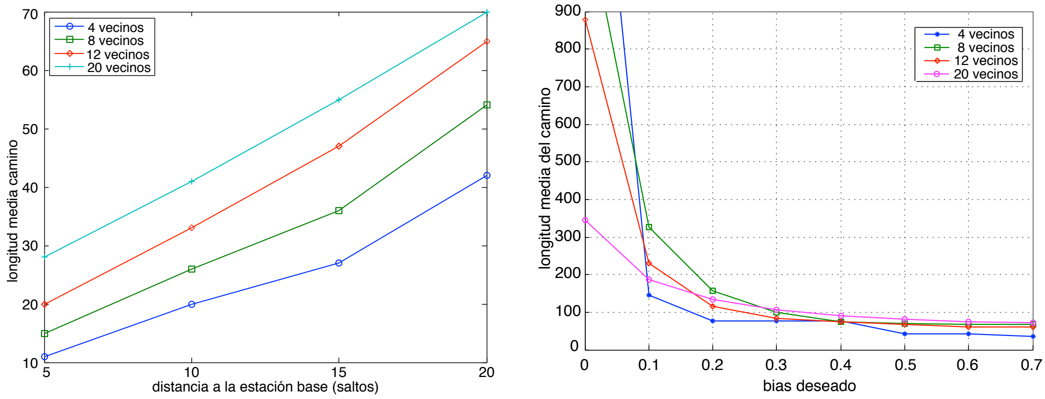
Algorithm 8 Algoritmo de perturbación

```
Input: $br \leftarrow \{L^{\mathcal{C}}, L^{\mathcal{E}}, L^{\mathcal{F}}\}$
Input: $sesgo$, $MAX\_ITER$
 1: $E \leftarrow energia(sesgo, br)$
 2: $i \leftarrow 0$
 3: while $(i < MAX\_ITER) \wedge (E \neq 0)$ do
 4:     $br' \leftarrow intercambiar(br)$
 5:     $E' \leftarrow energia(sesgo, br')$
 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 $br$
```

La principal ventaja de utilizar este tipo de estrategia frente a un algoritmo de búsqueda determinista se encuentra en el tiempo necesario para encontrar una solución (seudo-)óptima al problema, que dependiendo del tamaño del espacio de búsqueda puede diferir varios órdenes de magnitud. Sin embargo, su principal desventaja es que, al contrario que búsquedas deterministas, este tipo de algoritmos puede no encontrar la solución óptima al problema, aunque converge a ella. Nótese, que la perturbación introducida es difícilmente reversible si el valor de sesgo no es conocido, más aún cuando el algoritmo es no determinista.

A.5.3 Evaluación

La evaluación se ha realizado en nuestro propio simulador desarrollado en Matlab. Se han definido cuatro configuraciones de red en la que variamos el radio de transmisión para conseguir un número promedio de vecinos por nodo (4, 8, 12 y 20) diferente por cada configuración. Para ello, también debemos tener más o



(a) Numero esperado de saltos (b) Impacto del sesgo

Figure A.10: Tiempo de entrega de paquetes

menos nodos, conformando redes de 400, 1600, 1600 y 3600 para las configuraciones respectivas. Cada simulación cuenta con 500 pasos de simulación en los que se envían paquetes hacia la estación base. La evaluación se ha centrado en estudiar el nivel de protección de la solución frente a diferentes modelos de atacantes y la sobrecarga impuesta por la solución en función del tiempo promedio de entrega y las necesidades de tráfico falso.

En primer lugar hemos estudiado la sobrecarga que introduce nuestro protocolo de transmisión. Dada la naturaleza probabilística del protocolo de transmisión, los paquetes no siguen el camino óptimo a la estación base. En la Figura A.10, mostramos el número esperado de saltos para las cuatro configuraciones planteadas. En concreto, se presentan los resultados para nodos origen situados a diferentes distancias (5, 10, 15 y 20 saltos) de la estación base. Como era de esperar, a mayor distancia y mayor conectividad de los nodos, mayor es el número esperado de saltos. Sin embargo, es interesante observar que la velocidad de avance de los paquetes disminuye cuando los paquetes se acercan a su destino. Esto se debe a que en las proximidades de la estación base los nodos tienen un mayor número de vecinos $L^{\mathcal{F}}$.

En la Figura A.10b se muestra el impacto que tiene el algoritmo de perturbación sobre el tiempo de entrega. En este experimento todos los nodos están situados a distancia 20. Observamos que a medida que el sesgo se aproxima a cero el tiempo de entrega aumenta, siendo considerablemente mayor para configuraciones con un menor número de vecinos. Esto se debe a que las configuraciones con menos vecinos tienen menos formas de modificar las tablas de rutas. En

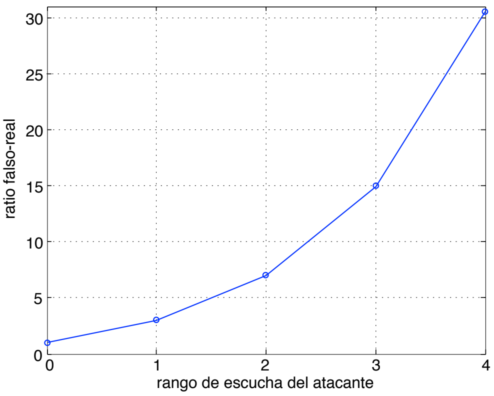
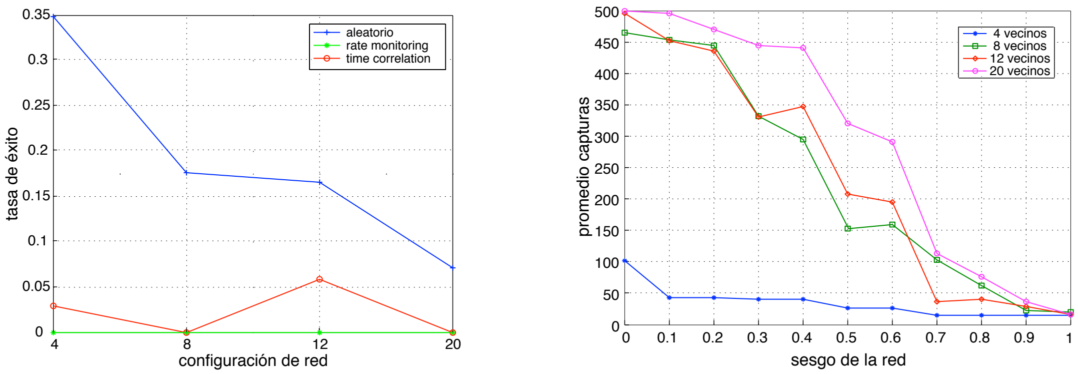


Figure A.11: Ratio de tráfico falso

concreto, cuando el sesgo deseado es cero, el sesgo promedio de la red para la configuración de cuatro vecinos es ligeramente inferior a cero, mientras que para la configuración de veinte vecinos el sesgo promedio está próximo a 0.1. En general, para un sesgo superior a 0.2 la longitud media de los caminos es inferior a 100 saltos.

En cuanto al tráfico falso, la tasa de inyección depende directamente del parámetro TTL_FALSO, cuyo valor depende del rango de escucha del atacante y que limita el número de paquetes falsos generados. En la Figure A.11 mostramos el ratio de mensaje falsos frente al tráfico real dependiendo del rango de escucha del atacante. Cuando el adversario sólo escucha los paquetes en su entorno inmediato, el ratio es 1 porque cada mensaje real va acompañado de un mensaje falso, que no vuelve a propagarse. A medida que el rango de escucha del adversario aumenta, el ratio lo hace en el orden de $\mathcal{O}(2^{n+1})$. A pesar de que se trata de una tasa exponencial, el modelo de atacante que consideramos es el típico de la literatura que tiene un rango de escucha local, similar al de un nodo ordinario. Basta con reenviar el tráfico falso una vez para evitar para evitar que el atacante observara todo el camino, lo cual es suficiente para evitar que obtenga información sensible.

Finalmente, estudiamos la robustez de nuestra solución frente a atacantes que realizan análisis de tráfico o captura de nodos. En la Figure A.12a se muestra como un modelo de atacante que se mueve de manera aleatoria, sin tener en cuenta las comunicaciones, tiene mayor probabilidad de llegar a la estación base que aquellos que recurren a técnicas de monitorización del tiempo y tasa de envío de paquetes. Además, como era de esperar, su tasa de éxito es mayor en configuraciones con un promedio de vecinos más bajo. Obsérvese que, del



(a) Ataques de análisis de tráfico

(b) Ataques de captura de nodos

Figure A.12: Tasa de éxito de diferentes adversarios

total de simulaciones lanzadas, el atacante que realiza monitorización de la tasa de envío nunca llega a la estación base mientras que el que realiza correlación de tiempos lo consigue en limitadas ocasiones. Esto se debe a que el atacante se encuentra inicialmente a distancia 5 y a la naturaleza de nuestro simulador, que es incapaz de determinar exactamente qué paquete es enviado antes. Por tanto, este atacante elige el siguiente salto de forma aleatoria entre los vecinos que envían mensajes.

En la Figure A.12b, el adversario comienza en un punto del extremo de la red y puede capturar hasta 500 nodos para llegar a la estación base. Además, asumimos que el atacante puede moverse al siguiente vecino tras obtener su identificador de la tabla de rutas aunque en un escenario real puede necesitar capturar a todos los vecinos del nodo para saber a cuál de ellos corresponde el identificador encontrado. La estrategia del atacante es moverse al primer nodo de la tabla de rutas que ha visitado un menor número de veces para evitar quedar atrapado en bucles. Los resultados muestran que, a medida que el sesgo de la red se acerca a cero, el adversario necesita capturar un mayor número de nodos para llegar a su destino. Sin embargo, un sesgo bajo influye negativamente en el tiempo de llegada de los paquetes a la estación base. En general, si consideramos que un atacante podría capturar hasta una décima parte de los nodos de la red, sería seguro utilizar un valor de sesgo menor o igual a 0.5. Por tanto, para garantizar un nivel de seguridad adecuado al tiempo que protegemos a la red de ataques de inspección de tablas de rutas el sesgo de la red debería encontrarse entre 0.2 y 0.5.

A.6 Conclusión

Esta tesis se ha concentrado en el problema de la privacidad de localización en WSNs. Este problema surge por la naturaleza inalámbrica de las comunicaciones, que permite a un observador analizar los patrones de tráfico y determinar la ubicación tanto de los nodos que generan eventos como de la estación base.

Dado que se trata de un problema de análisis de tráfico, el primer paso que consideramos fue estudiar si las soluciones de anonimato para redes de ordenadores eran aplicables a este entorno. Para ello, en lugar de limitarnos a estudiar los requisitos computacionales de estas soluciones, también prestamos atención a las características del modelo de comunicación, a los modelos de atacantes típicos y a las propiedades de anonimato que mejor se ajustaban al problema. Con todo esto concluimos que, a pesar de que puede parecer que los sistemas de comunicación anónima tradicionales son aplicables, estos no se ajustan a las necesidades del nuevo entorno. Por tanto, se hace necesario idear nuevas soluciones diseñadas específicamente para este redes de sensores.

Tras realizar un análisis exhaustivo del estado del arte y agrupar las diferentes soluciones en función del tipo de información que protegen y el modelo de atacante que consideraban, detectamos una serie de puntos de mejora que hemos plasmado en dos soluciones para hacer frente a atacantes con un rango de acción local. En primer lugar observamos que los mecanismos de protección diseñados hasta la fecha estaban activos en todo momento, independientemente de si el atacante se encuentra en el terreno analizando el tráfico. Esto nos llevó a plantearnos la pregunta de si era posible activar el mecanismo de protección sólo cuando fuera necesario, reduciendo así el retraso en las comunicaciones y el consumo energético. El mecanismo CALP da respuesta a esta pregunta, proporcionando un mecanismo de privacidad capaz de enviar los paquetes evitando la zona controlada por el atacante. Para ello, se aprovecha la capacidad sensorial de la red para detectar objetos en su entorno. A pesar de que la robustez de este esquema sólo fue estudiada para proteger la localización de nodos origen, creemos que además puede ser de gran utilidad para la protección de la estación base. La validez de esta afirmación la planteamos como trabajo futuro.

Por otra parte, observamos que las soluciones dedicadas a la protección de la estación base o bien eran muy costosas o bien no eran suficientemente robustas. Además, notamos que un atacante podría aprovechar las tablas de ruta de los nodos sensores para obtener información sobre la localización de la estación base. A pesar de esto, ningún trabajo hasta la fecha ha considerado este tipo de amenaza. Nuestra solución, HISP-NC, cuenta con dos mecanismos capaces de hacer frente a atacantes que realizan tanto ataques de análisis de tráfico como inspección de tablas de rutas. Durante la transmisión de datos, los paquetes de datos siguen un camino aleatorio guiado hacia la estación base, que se oculta enviando mensajes falsos. La idea es que cada vecino reciba en promedio la misma tasa de mensajes. Además, se ofrece un mecanismo que perturba las tablas de rutas de los nodos de manera que el atacante no pueda identificar con facilidad qué vecinos están más próximos a su objetivo. Uno de los principales puntos en contra de nuestra solución es que si el atacante tiene un rango de escucha amplio, se requiere una tasa de mensajes falsos que crece de manera exponencial. Encontrar la forma de reducir esta tasa se encuentra entre nuestras líneas de trabajo futuro.

A pesar del trabajo desarrollado durante esta tesis quedan muchos frentes de actuación por explorar. En primer lugar, creemos necesaria la búsqueda de nuevas soluciones que introduzcan métodos novedosos, como el propuesto por CALP, que permitan reducir el elevado coste de aplicar mecanismos de defensa. Otro aspecto de interés es el desarrollo de soluciones globales, capaces de proteger a los nodos origen y destino de manera simultánea. Asimismo, es fundamental el desarrollo de un marco unificado que permita cuantificar y comparar soluciones entre sí. En linea con esto último, es necesario definir modelos de atacante más inteligentes, capaces de adaptarse a la situación y utilizar información externa para alcanzar su objetivo. Por último, nos gustaría investigar y desarrollar soluciones de privacidad para entornos dinámicos, como la Internet de los Objetos, donde la movilidad es de vital importancia y los nodos no se comunican únicamente con la estación base sino con otros dispositivos. En este tipo de entornos no sólo aparecen nuevas formas de comunicación sino también nuevos modelos de atacante y, por tanto, será necesario desarrollar nuevas soluciones.

Bibliography

[1] Uday Acharya and Mohamed Younis. Increasing base-station anonymity in wireless sensor networks. *Ad Hoc Networks*, 8(8):791–809, 2010. ISSN 1570-8705. doi: DOI:10.1016/j.adhoc.2010.03.001.

[2] Isaac Agudo, Ruben Rios, and Javier Lopez. A Privacy-Aware Continuous Authentication Scheme for Proximity-Based Access Control. *Computers & Security*, 39, Part B:117–126, November 2013. ISSN 0167-4048. doi: 10.1016/j.cose.2013.05.004.

[3] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002. ISSN 0163-6804. doi: 10.1109/MCOM.2002.1024422.

[4] J.N. Al-Karaki and A.E. Kamal. Routing techniques in wireless sensor networks: a survey. *IEEE Wireless Communications*, 11(6):6 – 28, dec. 2004. ISSN 1536-1284. doi: 10.1109/MWC.2004.1368893.

[5] Alan F. Westin. *Privacy and Freedom.* New York Atheneum, 1 edition, 1967.

[6] Abdulrahman Alarifi and Wenliang Du. Diversify sensor nodes to improve resilience against node compromise. In *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks*, SASN '06, pages 101–112, New York, NY, USA, 2006. ACM. ISBN 1-59593-554-1. doi: 10.1145/1180345.1180359.

[7] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran. Statistical Framework for Source Anonymity in Sensor Networks. In *IEEE Global Telecommunications Conference*, GLOBECOM 2010, pages 1 –6, dec. 2010. doi: 10.1109/GLOCOM.2010.5684248.

[8] Basel Alomair, Andrew Clark, Jorge Cuellar, and Radha Poovendran. Towards a Statistical Framework for Source Anonymity in Sensor Networks. *IEEE Transactions on Mobile Computing*, 12(2):248 – 260, 2012. doi: 10.1109/TMC.2011.267.

[9] Nabil Ali Alrajeh, S. Khan, and Bilal Shams. Intrusion Detection Systems in Wireless Sensor Networks: A Review. *International Journal of Distributed Sensor Networks*, 2013(167575):7, 2013. doi: 10.1155/2013/167575.

[10] Anonymizer, Inc. Hide IP and Anonymous Web Browsing Software, May 2011. URL `http://www.anonymizer.com/`.

[11] Arduino. Lilypad arduino. Online, November 2014. `http://arduino.cc/en/pmwiki.php?n=Main/ArduinoBoardLilyPad`.

[12] Atmel. ATmega128, ATmega128L datasheet. Online. URL `http://www.atmel.com/Images/2467s.pdf`. Revised June 2011.

[13] Alexander Becher, Zinaida Benenson, and Maximillian Dornseif. Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks. In JohnA. Clark, RichardF. Paige, FionaA.C. Polack, and PhillipJ. Brooke, editors, *Security in Pervasive Computing*, volume 3934 of *Lecture Notes in Computer Science*, pages 104–118. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-33376-0. doi: 10.1007/11734666_9.

[14] Krista Bennett and Christian Grothoff. GAP - Practical Anonymous Networking. In Roger Dingledine, editor, *PET 2003*, volume 2760 of *LNCS*, pages 141–160, Dresden, Germany, 26–28 March 2003. Springer-Verlag.

[15] Krista Bennett, Christian Grothoff, Tzvetan Horozov, and J. T. Lindgren. An Encoding for Censorship-Resistant Sharing, 2003. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.2.800`.

[16] Alastair R. Beresford and Frank Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, January 2003. ISSN 1536-1268. doi: 10.1109/MPRV.2003.1186725.

[17] Shibasis Biswas, Sayan Mukherjee, and Krishnendu Mukhopadhyaya. A Countermeasure against Traffic-Analysis based Base Station Detection in

WSN, 2008. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.98.948`.

[18] Andrei Broder and Michael Mitzenmacher. Network applications of bloom filters: A survey. *Internet Mathematics*, 1(4):485–509, 2004. doi: 10.1080/15427951.2004.10129096.

[19] L. Buttyan, D. Gessner, A. Hessler, and Peter Langendoerfer. Application of wireless sensor networks in critical infrastructure protection: challenges and design options. *IEEE Wireless Communications*, 17(5):44–49, 2010. ISSN 1536-1284. doi: 10.1109/MWC.2010.5601957.

[20] Roy Campbell, Jalal Al-Muhtadi, Prasad Naldurg, Geetanjali Sampemane, and M. Mickunas. Towards Security and Privacy for Pervasive Computing. In Mitsuhiro Okada, BenjaminC. Pierce, Andre Scedrov, Hideyuki Tokuda, and Akinori Yonezawa, editors, *Software Security — Theories and Systems*, volume 2609 of *LNCS*, pages 1–15. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-00708-1. doi: 10.1007/3-540-36532-X_1.

[21] Bogdan Carbunar, Yang Yu, Weidong Shi, Michael Pearce, and Venu Vasudevan. Query privacy in wireless sensor networks. *ACM Trans. Sen. Netw.*, 6(2):14:1–14:34, March 2010. ISSN 1550-4859. doi: 10.1145/1689239.1689244.

[22] Claude Castelluccia, Aldar C-F. Chan, Einar Mykletun, and Gene Tsudik. Efficient and provably secure aggregation of encrypted data in wireless sensor networks. *ACM Trans. Sen. Netw.*, 5(3):20:1–20:36, June 2009. ISSN 1550-4859. doi: 10.1145/1525856.1525858.

[23] Matteo Ceriotti, Luca Mottola, Gian Pietro Picco, Amy L. Murphy, Stefan Guna, Michele Corra, Matteo Pozzi, Daniele Zonta, and Paolo Zanon. Monitoring heritage buildings with wireless sensor networks: The Torre Aquila deployment. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, IPSN '09, pages 277–288, Washington, DC, USA, 2009. IEEE Computer Society. ISBN 978-1-4244-5108-1. URL `http://dl.acm.org/citation.cfm?id=1602165.1602191`.

[24] Guofei Chai, Miao Xu, Wenyuan Xu, and Zhiyun Lin. Enhancing sink-location privacy in wireless sensor networks through k-anonymity. *International Journal of Distributed Sensor Networks*, 2012:16, 2012. doi: 10.1155/2012/648058.

[25] Shan Chang, Yong Qi, Hongzi Zhu, Mianxiong Dong, and Kaoru Ota. Maelstrom: Receiver-Location Preserving in Wireless Sensor Networks. In *Wireless Algorithms, Systems, and Applications*, volume 6843 of *LNCS*, pages 190–201. Springer, 2011. ISBN 978-3-642-23489-7. doi: 10.1007/978-3-642-23490-3_17.

[26] David Chaum. Untraceable Electronic Mail, Return addresses, and Digital Pseudonyms. *Commun. ACM*, 24(2):84–88, Feb. 1981.

[27] David Chaum. The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. *Journal of Cryptology*, 1:65–75, 1988.

[28] Omar Cheikhrouhou, Anis Koubaa, Manel Boujelben, and Mohamed Abid. A lightweight user authentication scheme for wireless sensor networks. In *Proceedings of the ACS/IEEE International Conference on Computer Systems and Applications*, pages 1–7, Hammamet, Tunisia, 16–19 May 2010. IEEE Computer Society, Washington, DC, USA. ISBN 978-1-4244-7716-6. doi: http://doi.ieeecomputersociety.org/10.1109/AICCSA.2010.5586995.

[29] Honglong Chen and Wei Lou. From Nowhere to Somewhere: Protecting End-to-End Location Privacy in Wireless Sensor Networks. In *29th International Performance Computing and Communications Conference*, IPCCC'10, pages 1–8. IEEE, 2010. doi: 10.1109/PCCC.2010.5682341.

[30] Juan Chen, Hongli Zhang, Binxing Fang, Xiaojiang Du, Lihua Yin, and Xiangzhan Yu. Towards Efficient Anonymous Communications in Sensor Networks. In *IEEE Global Telecommunications Conference*, GLOBECOM, pages 1–5, 2011. doi: 10.1109/GLOCOM.2011.6133560.

[31] Juan Chen, Xiaojiang Du, and Binxing Fang. An Efficient Anonymous Communication Protocol for Wireless Sensor Networks. *Wireless Communications and Mobile Computing*, 12(14):1302–1312, October 2012. ISSN 1530-8669. doi: 10.1002/wcm.1205.

[32] Xiangqian Chen, K. Makki, Kang Yen, and N. Pissinou. Node Compromise Modeling and its Applications in Sensor Networks. In *12th IEEE Symposium on Computers and Communications (ISCC 2007).*, pages 575 –582, july 2007. doi: 10.1109/ISCC.2007.4381514.

[33] Octav Chipara, Chenyang Lu, Thomas C. Bailey, and Gruia-Catalin Roman. Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, SenSys '10, pages 155–168, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0344-6. doi: 10.1145/1869983.1869999.

[34] Chi-Yin Chow, M.F. Mokbel, and Tian He. A Privacy-Preserving Location Monitoring System for Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 10(1):94 –107, January 2011. ISSN 1536-1233. doi: 10.1109/TMC.2010.145.

[35] William Conner, Tarek Abdelzaher, and Klara Nahrstedt. Using Data Aggregation to Prevent Traffic Analysis in Wireless Sensor Networks. In *Distributed Computing in Sensor Systems*, volume 4026 of *LNCS*, pages 202–217. Springer, 2006. ISBN 978-3-540-35227-3. doi: 10.1007/11776178_13.

[36] Jorge Cuéllar, Martín Ochoa, and Ruben Rios. Indistinguishable Regions in Geographic Privacy. In *Proceedings of the 2012 ACM Symposium on Applied Computing (SAC'12)*, pages 1463–1469, Riva del Garda (Trento), Italy, March 2012. ACM. ISBN 978-1-4503-0857-1. doi: 10.1145/2245276.2232010.

[37] Emiliano De Cristofaro, Xuhua Ding, and Gene Tsudik. Privacy-Preserving Querying in Sensor Networks. In *18th International Conference on Computer Communications and Networks*, ICCCN '09, pages 1–6, San Francisco, CA, 3-6 Aug. 2009. IEEE Computer Society, Washington, DC, USA. doi: 10.1109/ICCCN.2009.5235352.

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

[39] Jing Deng, Richard Han, and Shivakant Mishra. Enhancing Base Station Security in Wireless Sensor Networks. Technical Report CU-CS-951-03, University of Colorado, 2003. URL `http://www.cs.colorado.edu/~mishras/research/papers/tech03-1.pdf`.

[40] Jing Deng, Richard Han, and Shivakant Mishra. Countermeasures against traffic analysis attacks in wireless sensor networks. In *International Conference on Security and Privacy for Emerging Areas in Communications Networks*, SECURECOMM, pages 113–126, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2369-2. doi: 10.1109/SECURECOMM.2005.16.

[41] R. Di Pietro, L.V. Mancini, C. Soriente, A. Spognardi, and G. Tsudik. Data Security in Unattended Wireless Sensor Networks. *IEEE Transactions on Computers*, 58(11):1500 –1511, November 2009. ISSN 0018-9340. doi: 10.1109/TC.2009.109.

[42] Roberto Di Pietro and Alexandre Viejo. Location privacy and resilience in wireless sensor networks querying. *Comput. Commun.*, 34(3):515–523, March 2011. ISSN 0140-3664. doi: 10.1016/j.comcom.2010.05.014.

[43] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The Second-Generation Onion Router. In *13th conference on USENIX Security Symposium*, SSYM'04, pages 21–21, San Diego, CA, USA, 9–13 August 2004. USENIX Association, Berkeley, CA, USA.

[44] A. Eiben and J. Smith. *Introduction to Evolutionary Computing*. Natural Computing. Springer, 2 edition, 2007. ISBN 978-3-540-40184-1.

[45] A. El Kouche, L. Al-Awami, H. Hassanein, and K. Obaia. WSN application in the harsh industrial environment of the oil sands. In *7th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 613–618, 2011. doi: 10.1109/IWCMC.2011.5982603.

[46] H. Farhangi. The Path of the Smart Grid. *IEEE Power and Energy Magazine*, 8(1):18–28, Jan-Feb 2010. ISSN 1540-7977. doi: 10.1109/MPE.2009.934876.

[47] Emad Felemban. Advanced Border Intrusion Detection and Surveillance Using Wireless Sensor Network Technology. *International Journal of Communications, Network and System Sciences*, 6(5):251–259, 2013. doi: 10.4236/ijcns.2013.65028.

[48] Sharad Goel, Mark Robson, Milo Polte, and Emin Gün Sirer. Herbivore: A Scalable and Efficient Protocol for Anonymous Communication. Technical Report 2003-1890, Cornell University, Ithaca, NY, February 2003.

[49] C. Gómez, J. Paradells, and J. E. Caballero. *Sensors Everywhere: Wireless Network Technologies and Solutions*. Fundación Vodafone España, 2010. URL `http://fundacion.vodafone.es/static/fichero/pre_ucm_mgmt_002618.pdf`. ISBN 978-84-934740-5-8.

[50] Marco Gruteser, Graham Schelle, Ashish Jain, Rick Han, and Dirk Grunwald. Privacy-aware Location Sensor Networks. In *Proceedings of the 9th conference on Hot Topics in Operating Systems*, HOTOS'03, pages 28–28, Berkeley, CA, USA, 2003. USENIX Association. URL `http://www.usenix.org/events/hotos03/tech/full_papers/gruteser/gruteser.pdf`.

[51] Tian He, Sudha Krishnamurthy, Liqian Luo, Ting Yan, Lin Gu, Radu Stoleru, Gang Zhou, Qing Cao, Pascal Vicaire, John A. Stankovic, Tarek F. Abdelzaher, Jonathan Hui, and Bruce Krogh. VigilNet: An integrated sensor network system for energy-efficient surveillance. *ACM Transactions on Sensor Networks*, 2(1):1–38, February 2006. ISSN 1550-4859. doi: 10.1145/1138127.1138128.

[52] Wenbo He, Xue Liu, Hoang Nguyen, K. Nahrstedt, and T.T. Abdelzaher. PDA: Privacy-Preserving Data Aggregation in Wireless Sensor Networks. In *26th IEEE International Conference on Computer Communications*, pages 2045 –2053, May 2007. doi: 10.1109/INFCOM.2007.237.

[53] Jan Holvast. History of Privacy. In Vashek Matyás, Simone Fischer-Hübner, Daniel Cvrcek, and Petr Svenda, editors, *The Future of Identity in the Information Society*, volume 298 of *IFIP Advances in Information and Communication Technology*, pages 13–42. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-03314-8. doi: 10.1007/978-3-642-03315-5_2.

[54] James Horey, Michael M. Groat, Stephanie Forrest, and Fernando Esponda. Anonymous data collection in sensor networks. In *Proceedings of the 2007 Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking&Services (MobiQuitous)*, pages 1–8, Philadelphia, PA, USA, 6–10 August 2007. IEEE Computer Society, Washington, DC, USA. ISBN 978-1-4244-1024-8. doi: 10.1109/MOBIQ.2007.4451016.

[55] IEEE Smart Grid. Smart Grid Conceptual Model. Online, Sept 2013. URL `http://smartgrid.ieee.org/ieee-smart-grid/smart-grid-conceptual-model`.

[56] Arshad Jhumka, Matthew Leeke, and Sambid Shrestha. On the Use of Fake Sources for Source Location Privacy: Trade-Offs Between Energy and Privacy. *The Computer Journal*, 54(6):860–874, 2011. doi: 10.1093/comjnl/bxr010.

[57] Ying Jian, Shigang Chen, Zhan Zhang, and Liang Zhang. Protecting Receiver-Location Privacy in Wireless Sensor Networks. In *26th IEEE International Conference on Computer Communications*, INFOCOM, pages 1955–1963, May 2007. doi: 10.1109/INFCOM.2007.227.

[58] Ying Jian, Shigang Chen, Zhan Zhang, and Liang Zhang. A novel scheme for protecting receiver's location privacy in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 7(10):3769–3779, October 2008. ISSN 1536-1276. doi: 10.1109/T-WC.2008.070182.

[59] Jehn-Ruey Jiang, Jang-Ping Sheu, Ching Tu, and Jih-Wei Wu. An Anonymous Path Routing (APR) Protocol for Wireless Sensor Networks. *Journal of Information Science and Engineering*, 27(2):657–680, 2011.

[60] P. Kamat, Yanyong Zhang, W. Trappe, and C. Ozturk. Enhancing Source-Location Privacy in Sensor Network Routing. In *25th IEEE International Conference on Distributed Computing Systems*, ICDCS 2005, pages 599–608, June 2005. doi: 10.1109/ICDCS.2005.31.

[61] Pandurang Kamat, Wenyuan Xu, Wade Trappe, and Yanyong Zhang. Temporal Privacy in Wireless Sensor Networks. In *27th International Conference on Distributed Computing Systems*, ICDCS '07, page 23, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2837-3. doi: 10.1109/ICDCS.2007.146.

[62] Brad Karp and H. T. Kung. GPSR: Greedy Perimeter Stateless Routing for Wireless Networks. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 243–254, Boston, MA, USA, 6–11 August 2000. ACM, New York, NY, USA. ISBN 1-58113-197-6. doi: http://doi.acm.org/10.1145/345910.345953.

[63] L. Kazatzopoulos, C. Delakouridis, G. F. Marias, and P. Georgiadis. ihide: Hiding sources of information in wsns. In *International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing*, SecPerU'06, pages 41–48, Los Alamitos, CA, USA, 2006. IEEE Computer Society. ISBN 0-7695-2549-0. doi: 10.1109/SECPERU.2006.11.

[64] L. Kazatzopoulos, K. Delakouridis, and G. F. Marias. A privacy-aware overlay routing scheme in wsns. *Security and Communication Networks*, 4 (7):729–743, July 2011. ISSN 1939-0122. doi: 10.1002/sec.244.

[65] Sukun Kim, Shamim Pakzad, David Culler, James Demmel, Gregory Fenves, Steven Glaser, and Martin Turon. Health monitoring of civil infrastructures using wireless sensor networks. In *Proceedings of the 6th international conference on Information processing in sensor networks*, IPSN '07, pages 254–263, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-638-7. doi: 10.1145/1236360.1236395.

[66] Kae Hsiang Kwong, Tsung-Ta Wu, Hock Guan Goh, Konstantinos Sasloglou, Bruce Stephen, Ian Glover, Chong Shen, Wencai Du, Craig Michie, and Ivan Andonovic. Practical considerations for wireless sensor networks in cattle monitoring applications. *Computers and Electronics in Agriculture*, 81(0):33 – 44, 2012. ISSN 0168-1699. doi: 10.1016/j.compag.2011.10.013.

[67] Marc Langheinrich. A Privacy Awareness System for Ubiquitous Computing Environments. In Gaetano Borriello and LarsErik Holmquist, editors,

UbiComp 2002: Ubiquitous Computing, volume 2498 of *Lecture Notes in Computer Science*, pages 237–245. Springer Berlin Heidelberg, 2002. ISBN 978-3-540-44267-7. doi: 10.1007/3-540-45809-3_19.

[68] Rabia Latif and Mukhtar Hussain. Hardware-Based Random Number Generation in Wireless Sensor Networks. In *Advances in Information Security and Assurance*, volume 5576 of *LNCS*, pages 732–740. Springer, 2009. ISBN 978-3-642-02616-4. doi: 10.1007/978-3-642-02617-1_74.

[69] Loukas Lazos, Radha Poovendran, and James A. Ritcey. Analytic Evaluation of Target Detection in Heterogeneous Wireless Sensor Networks. *ACM Trans. Sen. Netw.*, 5(2):1–38, April 2009. ISSN 1550-4859. doi: http://doi.acm.org/10.1145/1498915.1498924.

[70] Brian Neil Levine and Clay Shields. Hordes: A Multicast Based Protocol for Anonymity. *J. Comput. Secur.*, 10(3):213–240, 2002. ISSN 0926-227X.

[71] Yun Li and Jian 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, pages 493–501, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-2907-3.

[72] Yun Li and Jian Ren. Providing Source-Location Privacy in Wireless Sensor Networks. In *4th International Conference on Wireless Algorithms, Systems, and Applications*, WASA '09, pages 338–347, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-03416-9. doi: 10.1007/978-3-642-03417-6_33.

[73] Yun Li, L. Lightfoot, and Jian Ren. Routing-Based Source-Location Privacy Protection in Wireless Sensor Networks. In *IEEE International Conference on Electro/Information Technology*, EIT'09, pages 29–34, 2009. doi: 10.1109/EIT.2009.5189579.

[74] Yun Li, Jian Ren, and Jie Wu. Quantitative Measurement and Design of Source-Location Privacy Schemes for Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 23:1302–1311, July 2012. doi: 10.1109/TPDS.2011.260.

[75] Leron Lightfoot, Yun Li, and Jian Ren. Preserving Source-Location Privacy in Wireless Sensor Network using STaR Routing. In *IEEE Global Telecommunications Conference*, GLOBECOM 2010, pages 1–5, 2010. doi: 10.1109/GLOCOM.2010.5683603.

[76] Leron Lightfoot, Yun Li, and Jian Ren. STaR: design and quantitative measurement of source-location privacy for wireless sensor networks. *Security and Communication Networks*, Online March 2012. ISSN 1939-0122. doi: 10.1002/sec.527.

[77] Benyuan Liu, Olivier Dousse, Jie Wang, and Anwar Saipulla. Strong Barrier Coverage of Wireless Sensor Networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, pages 411–420, Hong Kong, China, 27–30 May 2008. ACM New York, NY, USA. ISBN 978-1-60558-073-9. doi: http://doi.acm.org/10.1145/1374618.1374673. URL `http://doi.acm.org/10.1145/1374618.1374673`.

[78] Guojin Liu, Rui Tan, Ruogu Zhou, Guoliang Xing, Wen-Zhan Song, and Jonathan M. Lees. Volcanic earthquake timing using wireless sensor networks. In *Proceedings of the 12th international conference on Information processing in sensor networks*, IPSN '13, pages 91–102, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1959-1. doi: 10.1145/2461381.2461396.

[79] Giuseppe Lo Re, Fabrizio Milazzo, and Marco Ortolani. Secure random number generation in wireless sensor networks. In *Proceedings of the 4th international conference on Security of information and networks*, SIN '11, pages 175–182, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1020-8. doi: 10.1145/2070425.2070453.

[80] Javier Lopez, Rodrigo Roman, Isaac Agudo, and Carmen Fernandez-Gago. Trust Management Systems for Wireless Sensor Networks: Best practices. *Computer Communications*, 33(9):0140–3664, 2010. ISSN 0140-3664. doi: 10.1016/j.comcom.2010.02.006.

[81] Javier Lopez, Ruben Rios, and Jorge Cuellar. Preserving receiver-location privacy in wireless sensor networks. In Xinyi Huang and Jianying Zhou, editors, *Information Security Practice and Experience (ISPEC 2014)*, volume 8434 of *LNCS*, pages 15–27, Fuzhou (China), May 2014. Springer, Springer. doi: 10.1007/978-3-319-06320-1_3.

[82] Million Mafuta, Marco Zennaro, Antoine Bagula, Graham Ault, Harry Gombachika, , and Timothy Chadza. Successful Deployment of a Wireless Sensor Network for Precision Agriculture in Malawi. *International Journal of Distributed Sensor Networks*, 2013(150703):13, 2013. ISSN 1530-8677. doi: 10.1155/2013/150703.

[83] M. Mahmoud and X. Shen. A Cloud-Based Scheme for Protecting Source-Location Privacy against Hotspot-Locating Attack in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(10): 1805–1818, 2012. ISSN 1045-9219. doi: 10.1109/TPDS.2011.302.

[84] Mohamed Elsalih Mahmoud and Xuemin Shen. Secure and Efficient Source Location Privacy-Preserving Scheme for Wireless Sensor Networks. In *Proceedings of the IEEE International Conference on Communications, ICC'12*, pages 1123 – 1127, Ottawa, Canada, 10-15 June 2012. IEEE Communications Society. doi: 10.1109/ICC.2012.6363763.

[85] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, WSNA '02, pages 88–97, New York, NY, USA, 2002. ACM. ISBN 1-58113-589-0. doi: 10.1145/570738.570751.

[86] Guoqiang Mao, BarIs Fidan, and Brian D.O. Anderson. Wireless sensor network localization techniques. *Computer Networks*, 51(10): 2529 – 2553, 2007. ISSN 1389-1286. doi: DOI:10.1016/j.comnet.2006.11.018. URL http://www.sciencedirect.com/science/article/B6VRG-4MR88Y3-1/2/e1ec9202368c395d70268cd7e44b6484.

[87] Kirk Martinez and JaneK. Hart. Glacier Monitoring: Deploying Custom Hardware in Harsh Environments. In Elena Gaura, Michael Allen, Lewis Girod, James Brusey, and Geoffrey Challen, editors, *Wireless Sensor Networks*, pages 245–258. Springer US, 2010. ISBN 978-1-4419-5833-4. doi: 10.1007/978-1-4419-5834-1_9.

[88] Erika McCallister, Tim Grance, and Karen Scarfone. Guide to Protecting the Confidentiality of Personally Identifiable Information (PII). Special publication 800-122, National Institue of Standards and Technology

(NIST), 2010. URL http://csrc.nist.gov/publications/nistpubs/800-122/sp800-122.pdf.

[89] K. Mehta, Donggang Liu, and M. Wright. Location Privacy in Sensor Networks Against a Global Eavesdropper. In *IEEE International Conference on Network Protocols*, ICNP 2007, pages 314–323, Beijing, China, 16–19 Oct. 2007. IEEE. doi: 10.1109/ICNP.2007.4375862.

[90] Kiran Mehta, Donggang Liu, and Matthew Wright. Protecting Location Privacy in Sensor Networks Against a Global Eavesdropper. *IEEE Transactions on Mobile Computing*, 11(2):320–336, 2012. ISSN 1536-1233. doi: 10.1109/TMC.2011.32.

[91] MEMSIC. TelosB platform. Online, November 2014. http://www.memsic.com/wireless-sensor-networks/TPR2420.

[92] S. Misra and G. Xue. Efficient anonymity schemes for clustered wireless sensor networks. *International Journal of Sensor Networks*, 1(1):50–63, 2006.

[93] Alireza A. Nezhad, Dimitris Makrakis, and Ali Miri. Anonymous Topology Discovery for Multihop Wireless Sensor Networks. In *3rd ACM workshop on QoS and security for wireless and mobile networks*, Q2SWinet '07, pages 78–85, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-806-0. doi: 10.1145/1298239.1298254.

[94] Alireza A. Nezhad, Ali Miri, and Dimitris Makrakis. Location privacy and anonymity preserving routing for wireless sensor networks. *Computer Networks*, 52(18):3433 – 3452, Dec. 2008. ISSN 1389-1286.

[95] Dragos Niculescu and Badri Nath. Trajectory Based Forwarding and Its Applications. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 260–272, San Diego, CA, USA, 14–19 September 2003. ACM, New York, NY, USA. ISBN 1-58113-753-2. doi: http://doi.acm.org/10.1145/938985.939012.

[96] Oracle Labs. Sun spot world. Online, November 2014. http://www.sunspotworld.com/.

[97] Stefano Ortolani, Mauro Conti, Bruno Crispo, and Roberto Di Pietro. Events privacy in WSNs: A new model and its applications. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1 –9, june 2011. doi: 10.1109/WoWMoM.2011.5986491.

[98] Yi Ouyang, Zhengyi Le, Guanling Chen, James Ford, and Fillia Makedon. Entrapping Adversaries for Source Protection in Sensor Networks. In *2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, WOWMOM '06, pages 23–34, Washington, DC, USA, 2006. IEEE Computer Society. ISBN 0-7695-2593-8. doi: 10.1109/WOWMOM.2006.40.

[99] Yi Ouyang, Zhengyi Le, Yurong Xu, N. Triandopoulos, Sheng Zhang, J. Ford, and F. Makedon. Providing Anonymity in Wireless Sensor Networks. In *IEEE International Conference on Pervasive Services*, pages 145–148, July 2007. doi: 10.1109/PERSER.2007.4283904.

[100] Celal Ozturk, Yanyong Zhang, and Wade Trappe. Source-Location Privacy in Energy-Constrained Sensor Network Routing. In *2nd ACM workshop on Security of ad hoc and sensor networks*, SASN '04, pages 88–93, Washington, DC, USA, 2004. ACM New York, NY, USA. ISBN 1-58113-972-1. doi: 10.1145/1029102.1029117.

[101] S. Pai, S. Bermudez, S.B. Wicker, M. Meingast, T. Roosta, S. Sastry, and D.K. Mulligan. Transactional Confidentiality in Sensor Networks. *IEEE Security & Privacy*, 6(4):28–35, July-Aug. 2008. ISSN 1540-7993. doi: 10.1109/MSP.2008.107.

[102] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, Aug. 2010. URL `http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf`. v0.34.

[103] K. Pongaliur and Li Xiao. Maintaining source privacy under eavesdropping and node compromise attacks. In *IEEE International Conference on Computer Communications*, INFOCOM, pages 1656 –1664, april 2011. doi: 10.1109/INFCOM.2011.5934959.

[104] Kanthakumar Pongaliur and Li Xiao. Sensor Node Source Privacy and Packet Recovery Under Eavesdropping and Node Compromise Attacks. *ACM Transactions on Sensor Networks*, 9(4):50:1–50:26, July 2013. ISSN 1550-4859. doi: 10.1145/2489253.2489267.

[105] Alejandro Proano and Loukas Lazos. Hiding contextual information in wsns. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, WoWMoM, pages 1 –6, june 2012. doi: 10.1109/WoWMoM.2012.6263769.

[106] Alejandro Proano and Loukas Lazos. Perfect Contextual Information Privacy in WSNs under Colluding Eavesdroppers. In *6th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec'13, Budapest, Hungary, April 17-19 2013. ACM.

[107] Jean-François Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 10–29. Springer-Verlag, LNCS 2009, July 2000.

[108] M.G. Reed, P.F. Syverson, and D.M. Goldschlag. Anonymous Connections and Onion Routing. *IEEE Journal on Selected Areas in Communications*, 16(4):482–494, May 1998. ISSN 0733-8716. doi: 10.1109/49.668972.

[109] M.K. Reiter and A.D. Rubin. Crowds: Anonymity for Web Transactions. *ACM transactions on information and system security*, 1(1):66–92, 1998.

[110] Jian Ren, Yun Li, and Tongtong Li. Routing-Based Source-Location Privacy in Wireless Sensor Networks. In *IEEE International Conference on Communications*, ICC'09, pages 620–624, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-3434-3. doi: 10.1109/ICC.2009.5199430.

[111] Ruben Rios and Javier Lopez. Source Location Privacy Considerations in Wireless Sensor Networks. In Lidia Fuentes, Nadia Gámez, and José Bravo, editors, *4th International Symposium of Ubiquitous Computing and Ambient Intelligence (UCAmI'10)*, pages 29 – 38, Valencia (Spain), Sept. 2010. ISBN 978-84-92812-61-5.

[112] Ruben Rios and Javier Lopez. Analysis of location privacy solutions in wireless sensor networks. *IET Communications*, 5:2518 – 2532, 2011. ISSN 1751-8628. doi: 10.1049/iet-com.2010.0825. Impact Factor: 0.83.

[113] Ruben Rios and Javier Lopez. Exploiting Context-Awareness to Enhance Source-Location Privacy in Wireless Sensor Networks. *The Computer Journal*, 54(10):1603–1615, 2011. doi: 10.1093/comjnl/BXR055. Impact Factor: 0.79.

[114] Ruben Rios and Javier. Lopez. Adecuación de soluciones de anonimato al problema de la privacidad de localización en WSNs. In R. Uribeetxeberria U. Zurutuza and I. Arenaza-Nuño, editors, *XII Reunión Española sobre Criptología y Seguridad de la Información (RECSI 2012)*, pages 309–314, San Sebastian (Spain), Sept. 2012.

[115] Ruben Rios and Javier Lopez. (Un)Suitability of Anonymous Communication Systems to WSN. *IEEE Systems Journal*, 7(2):298 – 310, June 2013. ISSN 1932-8184. doi: 10.1109/JSYST.2012.2221956. Impact Factor: 1.27.

[116] Ruben Rios, Isaac Agudo, and Jose L. Gonzalez. Implementación de un esquema de localización privada y segura para interiores. In Yannis Dimitriadis and María Jesús Verdú Pérez, editors, *IX Jornadas de Ingeniería Telemática (JITEL'10)*, pages 237 – 244, Valladolid (Spain), Sept. 2010. ISBN 978-84-693-5398-1.

[117] Ruben Rios, Jorge Cuellar, and Javier Lopez. Robust Probabilistic Fake Packet Injection for Receiver-Location Privacy in WSN. In S. Foresti, M. Yung, and F. Martinelli, editors, *17th European Symposium on Research in Computer Security (ESORICS 2012)*, volume 7459 of *LNCS*, pages 163–180, Pisa (Italy), Sept. 2012. Springer. doi: 10.1007/978-3-642-33167-1_10.

[118] Ruben Rios, Jose A Onieva, and Javier Lopez. HIDE_DHCP: Covert Communications Through Network Configuration Messages. In *Proceedings of the 27th IFIP TC 11 International Information Security Conference*, volume 376 of *IFIP AICT*, pages 162–173. Springer, Heraklion, Crete (Greece), June 2012. doi: 10.1007/978-3-642-30436-1_14.

[119] Ruben Rios, Jorge Cuellar, and Javier Lopez. Ocultación de la estación base en redes inalámbricas de sensores. In Jesús E. Díaz Verdejo, Jorge Navarro

Ortiz, and Juan J. Ramos Muñoz, editors, *XI Jornadas de Ingeniería Telemática (JITEL 2013)*, pages 481–486, Granada, Oct 2013 2013. Asociación Telemática. ISBN 978-84-616-5597-7.

[120] Ruben Rios, Jose A. Onieva, and Javier Lopez. Covert Communications through Network Configuration Messages. *Computers & Security*, 39, Part A:34–46, 2013. ISSN 0167-4048. doi: 10.1016/j.cose.2013.03.004.

[121] Ruben Rios, Javier Lopez, and Jorge Cuellar. Location Privacy in WSNs: Solutions, Challenges, and Future Trends. In *Foundations of Security Analysis and Design (FOSAD) VII*, volume 8604, pages 244–282. Springer, 2014. doi: 10.1007/978-3-319-10082-1_9.

[122] Ruben Rios, Jorge Cuellar, and Javier Lopez. Probabilistic receiver-location privacy protection in wireless sensor networks. *Information Sciences*, Accepted for publication. Impact Factor: 3.89.

[123] Rodrigo Roman. *Application-Driven Security in Wireless Sensor Networks.* Phd thesis, University of Malaga, 2008.

[124] R.A. Shaikh, H. Jameel, B.J. d'Auriol, Sungyoung Lee, Young-Jae Song, and Heejo Lee. Network Level Privacy for Wireless Sensor Networks. In *4th International Conference on Information Assurance and Security*, ISIAS '08, pages 261–266, Sept. 2008. doi: 10.1109/IAS.2008.36.

[125] S. Shakkottai. Asymptotics of query strategies over a sensor network. In *23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1 of *INFOCOM 2004*, pages 548–557, 2004. doi: 10.1109/INFCOM.2004.1354526.

[126] E.M. Shakshuki, T.R. Sheltami, Nan Kang, and Xinyu Xing. Tracking Anonymous Sinks in Wireless Sensor Networks. In *International Conference on Advanced Information Networking and Applications*, AINA, pages 510–516. IEEE Computer Society, May 2009. doi: 10.1109/AINA.2009.61.

[127] Claude Elwood Shannon. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, 28:656–715, 1949. URL `http://netlab.cs.ucla.edu/wiki/files/shannon1949.pdf`.

[128] M. Shao, W. Hu, S. Zhu, G. Cao, S. Krishnamurthy, and T. La Porta. Cross-layer Enhanced Source Location Privacy in Sensor Networks. In *IEEE Conference on Sensor, Mesh, and Ad Hoc Communications and Networks*, SECON '09, pages 1–9. IEEE Communications Society, June 2009.

[129] Min Shao, Yi Yang, Sencun Zhu, and Guohong Cao. Towards Statistically Strong Source Anonymity for Sensor Networks. In *27th IEEE Conference on Computer Communications*, INFOCOM 2008, pages 466–474, April 2008. doi: 10.1109/INFOCOM.2008.19.

[130] Min Shao, Sencun Zhu, Wensheng Zhang, Guohong Cao, and Yi Yang. pdcs: Security and privacy support for data-centric sensor networks. *Mobile Computing, IEEE Transactions on*, 8(8):1023–1038, Aug. 2009. ISSN 1536-1233. doi: 10.1109/TMC.2008.168.

[131] Jang-Ping Sheu, Jehn-Ruey Jiang, and Ching Tu. Anonymous Path Routing in Wireless Sensor Networks. In *IEEE International Conference on Communications*, ICC '08, pages 2728 –2734, May 2008. doi: 10.1109/ICC.2008.515.

[132] Victor Shnayder, Mark Hempstead, Bor-rong Chen, Geoff Werner Allen, and Matt Welsh. Simulating the power consumption of large-scale sensor network applications. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 188–200, New York, NY, USA, 2004. ACM. ISBN 1-58113-879-2. doi: 10.1145/1031495.1031518.

[133] S. Spiekermann and L.F. Cranor. Engineering Privacy. *IEEE Transactions on Software Engineering*, 35(1):67–82, January 2009. ISSN 0098-5589. doi: 10.1109/TSE.2008.88.

[134] Texas Instruments. Datasheet MSP430F15x, MSP430F16x, MSP430F161x mixed signal microcontroller (rev. G). Online, October 2002. URL `http://www.ti.com/lit/ds/symlink/msp430f1611.pdf`. Revised March 2011.

[135] The Institute of Electrical and Electronics Engineers (IEEE). IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal

Area Networks (LR-WPANs), 2003. URL `http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf`.

[136] Binod Vaidya, Min Chen, and Joel J. P. C. Rodrigues. Improved robust user authentication scheme for wireless sensor networks. In *IEEE Conference on Wireless Communication and Sensor Networks*, pages 1 – 6, Allahabad, India, 15-19 Dec 2009. IEEE Xplore. doi: 10.1109/WCSN.2009.5434810.

[137] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer, and P. Doody. Internet of Things Strategic Research Roadmap. Technical report, Cluster of European Research Projects on the Internet of Things (CERP-IoT), 2011. URL `http://www.internet-of-things-research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf`.

[138] Tuan Manh Vu, Reihaneh Safavi-Naini, and Carey 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, pages 112–123, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-936-7. doi: 10.1145/1755688.1755703.

[139] J.P. Walters, Z. Liang, W. Shi, and V. Chaudhary. *Security in Distributed, Grid, and Pervasive Computing*, chapter Wireless Sensor Network Security: A Survey, pages 367–409. Auerbach Pub, 2007.

[140] Haodong Wang, Bo Sheng, and Qun Li. Privacy-aware routing in sensor networks. *Computer Networks*, 53(9):1512–1529, 2009. ISSN 1389-1286. doi: 10.1016/j.comnet.2009.02.002.

[141] Hou-Jie Wang and Tien-Ruey Hsiang. Defending Traffic Analysis with Communication Cycles in Wireless Sensor Networks. In *10th International Symposium on Pervasive Systems, Algorithms, and Networks*, ISPAN, pages 166 –171, 2009. doi: 10.1109/I-SPAN.2009.78.

[142] Yong Wang, G. Attebury, and B. Ramamurthy. A survey of security issues in wireless sensor networks. *IEEE Communications Surveys Tutorials*, 8 (2):2–23, 2006. ISSN 1553-877X. doi: 10.1109/COMST.2006.315852.

[143] Samuel Warren and Louis Brandeis. The Right to Privacy. *Harvard Law Review*, IV(5), December 1890. URL `http://groups.csail.mit.edu/mac/classes/6.805/articles/privacy/Privacy_brand_warr2.html`.

[144] Wang Wei-Ping, Chen Liang, and Wang Jian-Xin. A source-location privacy protocol in WSN based on locational angle. In *IEEE International Conference on Communications*, ICC '08, pages 1630–1634, Beijing, 19–23 May 2008. IEEE Communications Society. doi: 10.1109/ICC.2008.315.

[145] Mark Weiser. The Computer for the 21st Century. *Scientific American*, 265:94–104, 1991. doi: 10.1038/scientificamerican0991-94. URL `http://wiki.daimi.au.dk/pca/_files/weiser-orig.pdf`.

[146] J. Wilson and N. Patwari. Radio Tomographic Imaging with Wireless Networks. *IEEE Transactions on Mobile Computing*, 9(5):621 –632, May 2010. ISSN 1536-1233. doi: 10.1109/TMC.2009.174.

[147] A. Wood, J.A. Stankovic, G. Virone, L. Selavo, Zhimin He, Qiuhua Cao, Thao Doan, Yafeng Wu, Lei Fang, and R. Stoleru. Context-aware wireless sensor networks for assisted living and residential monitoring. *IEEE Network*, 22(4):26–33, 2008. ISSN 0890-8044. doi: 10.1109/MNET.2008.4579768.

[148] Yong Xi, L. Schwiebert, and Weisong Shi. Preserving Source Location Privacy in Monitoring-Based Wireless Sensor Networks. In *20th International Parallel and Distributed Processing Symposium*, IPDPS 2006, page 8 pp., April 2006. doi: 10.1109/IPDPS.2006.1639682.

[149] Yi Yang, Min Shao, Sencun Zhu, Bhuvan Urgaonkar, and Guohong Cao. Towards Event Source Unobservability with Minimum Network Traffic in Sensor Networks. In *1st ACM conference on Wireless network security*, WiSec'08, pages 77–88, New York, NY, USA, 2008. ACM. ISBN 978-1-59593-814-5. doi: 10.1145/1352533.1352547.

[150] Yi Yang, Sencun Zhu, Guohong Cao, and Thomas LaPorta. An Active Global Attack Model for Sensor Source Location Privacy: Analysis

and Countermeasures. In *Security and Privacy in Communication Networks*, volume 19 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 373–393. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-05284-2. doi: 10.1007/978-3-642-05284-2_22.

[151] Jianbo Yao. Source-location privacy based on directed greedy walk in wireless sensor networks. In *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on*, pages 1–4, sept. 2010. doi: 10.1109/WICOM.2010.5601128.

[152] Jianbo Yao and Guangjun Wen. Preserving source-location privacy in energy-constrained wireless sensor networks. In *Proceedings of the 28th International Conference on Distributed Computing Systems Workshops*, ICDCSW '08, pages 412–416, Washington, DC, USA, 2008. IEEE Computer Society. ISBN 978-0-7695-3173-1. doi: 10.1109/ICDCS.Workshops.2008.42.

[153] Lin Yao, Lin Kang, Pengfei Shang, and Guowei Wu. Protecting the sink location privacy in wireless sensor networks. *Personal and Ubiquitous Computing*, pages 1–11, 2012. ISSN 1617-4909. doi: 10.1007/s00779-012-0539-9. 10.1007/s00779-012-0539-9.

[154] Lin Yao, Lin Kang, Fangyu Deng, Jing Deng, and Guowei Wu. Protecting source–location privacy based on multirings in wireless sensor networks. *Concurrency and Computation: Practice and Experience*, Online June 2013. ISSN 1532-0634. doi: 10.1002/cpe.3075.

[155] Bidi Ying, Jose R. Gallardo, Dimitrios Makrakis, and Hussein T. Mouftah. Concealing of the Sink Location in WSNs by Artificially Homogenizing Traffic Intensity. In *The First International Workshop on Security in Computers, Networking and Communications (INFOCOM Workshops)*, pages 988 – 993, April 2011. doi: 10.1109/INFCOMW.2011.5928957.

[156] Bidi Ying, D. Makrakis, and H.T. Mouftah. A Protocol for Sink Location Privacy Protection in Wireless Sensor Networks. In *IEEE Global Telecommunications Conference*, GLOBECOM, pages 1 –5, Houston, TX, USA, 5-9 Dec. 2011. IEEE Communications Society. doi: 10.1109/GLOCOM.2011.6133922.

[157] Dian Zhang, Jian Ma, Quanbin Chen, and Lionel M. Ni. An RF-Based System for Tracking Transceiver-free Objects. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications*, pages 135 –144, White Plains, New York, USA, 19-23 March 2007. IEEE Computer Society, Washington, DC, USA. doi: 10.1109/PERCOM.2007.8.

[158] Junqi Zhang and Vijay Varadharajan. Wireless sensor network key management survey and taxonomy. *Journal of Network and Computer Applications*, 33(2):63 – 75, 2010. ISSN 1084-8045. doi: 10.1016/j.jnca.2009.10.001.

[159] Lei Zhang, Honggang Zhang, Mauro Conti, Roberto Di Pietro, Sushil Jajodia, and LuigiVincenzo Mancini. Preserving privacy against external and internal threats in WSN data aggregation. *Telecommunication Systems*, 52 (4):2163–2176, 2013. ISSN 1018-4864. doi: 10.1007/s11235-011-9539-8.

[160] Wensheng Zhang, Chuang Wang, and Taiming Feng. GP^2S: Generic Privacy-Preservation Solutions for Approximate Aggregation of Sensor Datas. In *Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, PerCom'08, pages 179–184, Hong Kong, China, 17-21 March 2008. IEEE Computer Society, Washington, DC, USA. doi: 10.1109/PERCOM.2008.60.