

Data-level privacy through data perturbation in distributed multi-application environments



Tulio de Souza
Corpus Christi College
University of Oxford

A dissertation submitted for the degree of
Doctor of Philosophy

April 2016

Abstract

Wireless sensor networks used to have a main role as a monitoring tool for environmental purposes and animal tracking. This spectrum of applications, however, has dramatically grown in the past few years. Such evolution means that what used to be application-specific networks are now multi application environments, often with federation capabilities. This shift results in a challenging environment for data privacy, mainly caused by the broadening of the spectrum of data access points and involved entities.

This thesis first evaluates existing privacy preserving data aggregation techniques to determine how suitable they are for providing data privacy in this more elaborate environment. Such evaluation led to the design of the set difference attack, which explores the fact that they all rely purely on data aggregation to achieve privacy, which is shown through simulation not to be suitable to the task. It also indicates that some form of uncertainty is required in order to mitigate the attack. Another relevant finding is that the attack can also be effective against standalone networks, by exploring the node availability factor.

Uncertainty is achieved via the use of differential privacy, which offers a strong and formal privacy guarantee through data perturbation. In order to make it suitable to work in a wireless sensor network environment, which mainly deals with time-series data, two new approaches to address it have been proposed. These have a contrasting effect when it comes to utility and privacy levels, offering a flexible balance between privacy and data utility for sensed entities and data analysts/consumers.

Lastly, this thesis proposes a framework to assist in the design of privacy preserving data aggregation protocols to suit application needs while at the same time complying with desired privacy requirements. The framework's evaluation compares and contrasts several scenarios to demonstrate

the level of flexibility and effectiveness that the designed protocols can provide.

Overall, this thesis demonstrates that data perturbation can be made significantly practical through the proposed framework. Although some problems remain, with further improvements to data correlation methods and better use of some intrinsic characteristics of such networks, the use of data perturbation may become a practical and efficient privacy preserving mechanism for wireless sensor networks.

Acknowledgements

Firstly I would like to thank my supervisors, Andrew Martin and Ian Brown, who have provided endless insight, support and encouragement throughout my DPhil. I am grateful for the opportunities and ideas that they have provided.

I also thank my wife Giani, son Lucca and daughter Laura for supporting me throughout my quest for a DPhil. When it started, we were only my wife and me. As I approach the end of this journey, the family is twice as big. I could not be more grateful to Giani for being part of it.

I thank my parents for giving me the opportunity to chase my dreams and for their understanding and advice. I would not be here without their help and guidance over the last 31 years.

Throughout my DPhil I have been fortunate to study with brilliant people. In particular, I thank John, Cornelius, Shamal, Ronald, Joe, Ivan, Anbang, Hongkai and the rest of the Security Reading Group. Our enjoyable weekly discussions have contributed to this dissertation as well as my education in computer security. I am also extremely grateful to the Oxford University Computing Laboratory and Software Engineering Programme for providing several excellent courses and opportunities throughout my studies. I thank my examiners for their useful feedback and suggestions. A special thanks goes to Joss Wright for countless discussion and his incredible cleverness in a varying range of subjects. His support has assisted me in shaping up this thesis.

My friends deserve enormous credit for making my studies such good fun. Special thanks goes to Jean, Marcelo, Felipe, Juliano, Fernando and Irfan. I am grateful to Rojas Haleem for proofreading this dissertation, as well as all his always helpful mentality. I also thank my other friends whom deserve a mention but will, unfortunately, not get one.

This dissertation was mainly funded by the EPSRC, which I owe a debt of gratitude for making this project possible.

Contents

1	Introduction	1
1.1	Why Does Data Privacy Matter in Wireless Sensor Networks?	4
1.1.1	Scenario Shifting	4
1.1.2	Smart cities	5
1.1.3	Infrastructure Monitoring (Case Study)	6
1.2	Challenges	7
1.3	Contributions and Dissertation Structure	8
2	Wireless Sensor Network	11
2.1	Basic Architecture and Characteristics	12
2.2	Organisational Structure	12
2.2.1	Standalone Networks	12
2.2.2	Multi-Applications Networks	13
2.2.3	Federated Multi-Application Networks	13
2.3	Middleware	13
2.4	Data Aggregation	14
2.5	Privacy in WSN	16
2.5.1	Contextual Privacy	16
2.5.2	Data Privacy	16
2.6	Privacy-Preserving Data Aggregation Protocols in WSN	18
2.6.1	Goals	18
2.6.2	Clustering	18
2.6.3	Slicing	19
2.6.4	Privacy Homomorphisms	20
2.7	Data-level privacy	21
2.7.1	Models for Privacy Mechanisms	22
2.7.1.1	Non-interactive	22
2.7.1.2	Interactive	22

2.8	Differential Privacy	23
2.8.1	PINQ	24
2.8.2	PASTE	25
2.9	Summary	26
3	Set Difference Attack	28
3.1	System and Attacker Model	29
3.2	The attack	30
3.3	Set Difference Attacks in Detail	31
3.3.1	Isolated Cluster	32
3.3.2	Combined Subsets	32
3.3.3	Total Set Coverage	33
3.3.4	Attack Recursion	33
3.4	Attacking Existing Protocols	34
3.4.1	Node Availability	34
3.5	Feasibility of Set Difference Attacks	35
3.6	Preventing Set Difference Attacks	37
3.6.1	A Note on Fixed Clustering	37
3.6.2	Data Perturbation	38
3.6.2.1	Perturbation alongside other mechanisms	39
3.7	Conclusion	39
4	Formally Perturbing Wireless Sensor Networks	42
4.1	Misalignment of Granularity	43
4.1.1	Misalignment in Practice	44
4.2	Statistical Set Difference Attack	46
4.3	Mitigation	47
4.3.1	A Note on Query Limitation	47
4.3.2	Spatially-limited Data Grouping	48
4.3.3	Enhanced Budget Management	49
4.4	Evaluation	50
4.5	Conclusion	52

5	Framework for Designing Privacy-preserving Data Aggregation Protocols	55
5.1	Placing the Trust	56
5.2	Network Topology	57
5.3	Storage and Data Feeding Model	58
5.4	Application Needs vs Privacy Goals	58
5.4.1	Data Grouping	59
5.4.2	Budget Management	59
5.5	Data Perturbation in Conjunction with Other Techniques	60
5.5.1	Clustering	60
5.5.2	Slicing	61
5.5.3	Other Techniques	61
5.6	Framework Overview	61
5.7	Conclusion	64
6	Evaluation	66
6.1	Approach	66
6.2	Centralised Trust Model	67
6.3	Decentralised Trust Model	68
6.4	Hybrid Trust Model	73
6.5	Conclusion	75
7	Conclusion and Future Work	78
7.1	Contributions	78
7.1.1	Unsuitability of Existing Privacy-preserving Protocols in Multi-application Wireless Sensor Networks	78
7.1.2	Formally Achieving Privacy in Wireless Sensor Networks	79
7.1.3	Framework for Designing Privacy-preserving Data Aggregation Protocols	79
7.2	Future Work	80
7.2.1	Enhancement of the Framework	80
7.2.2	Improving Data Utility	81
7.2.3	Privacy by Design	81
7.2.4	Mobile Network	82
7.3	Summary	82
	Bibliography	84

List of Figures

2.1	Private clustering in WSNs	19
2.2	Private slicing showing the path of two private data ‘slices’ travelling across the network from f to a	20
2.3	Homomorphic encryption in WSNs. Values are aggregated in encrypted form at each node.	21
2.4	Overlapping Laplace distributions, means μ_1 and μ_2 , showing comparative probabilities of two values, a and b , drawn according to either distribution.	24
3.1	Simple set differences in a WSN.	31
3.2	A set difference attack combining multiple disjoint subsets.	32
3.3	Mean average and sample standard deviation of randomly-chosen sets required in networks of varying size before a successful set difference attack.	36
3.4	Mean standard error (MSE) for various values of σ as application size increases.	39
4.1	It shows the average desk occupancy over a two-hour period. The occupancy results are presented in its accurate value and the standard deviation resulting of 500 runs of the solution proposed here. These results vary accordingly to the size of the network and they are based on the data centrally located at the sink.	51
4.2	It shows the average desk occupancy over a two-hour period. The occupancy results are presented in its accurate value and the standard deviation resulting of 500 runs of the solution proposed here. These results vary accordingly to the size of the network and they are based on the data being fully distributed (only the node themselves have access to the raw data).	52

4.3	It shows the average desk occupancy over a two-hour period based of 15 minutes sampling. The occupancy results are presented in its accurate value (based on all 120 entries for each user) and the standard deviation resulting of 500 runs of our solution using 15-minute sampling. These results vary accordingly to the size of the network and they are based on the data being fully distributed (only the node themselves have access to the raw data).	53
6.1	It presents the behaviour of a network that adopts a centralised trust model with budget expenditure 5.	68
6.2	It presents the behaviour of a network that adopts a centralised trust model with budget expenditure 1.	68
6.3	It presents the behaviour of a network that adopts a centralised trust model with budget expenditure 0.2.	69
6.4	It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 5.	69
6.5	It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 1.	70
6.6	It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 0.5.	70
6.7	It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 5. The questions cover 4 times more data points than the size of the data grouping in use. For this particular experiment it used 240 data points from each node since the data grouping size is 60 data points.	71
6.8	It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 1. The questions cover 4 times more data points than the size of the data grouping in use. For this particular experiment it used 240 data points from each node since the data grouping size is 60 data points.	72
6.9	It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 0.5. The questions cover 4 times more data points than the size of the data grouping in use. For this particular experiment it used 240 data points from each node since the data grouping size is 60 data points.	72

6.10	It presents the behaviour of a network that adopted a hybrid trust model with 10 clusters of nodes and budget expenditure 5.	73
6.11	It presents the behaviour of a network that adopted a hybrid trust model with 10 clusters of nodes and budget expenditure 1.	74
6.12	It presents the behaviour of a network that adopted a hybrid trust model with 10 clusters of nodes and budget expenditure 0.2.	74
6.13	It compares the behaviour of the network that adopted a hybrid trust model, from the point of view of data utility, for a total of 10 clusters.	75
6.14	It compares the behaviour of the network that adopted a hybrid trust model, from the point of view of data utility, for a total of 50 clusters.	75
6.15	It compares the behaviour of the network that adopted a hybrid trust model, from the point of view of data utility, for a total of 100 clusters.	76

List of Tables

4.1	This table represents a sample of the occupancy table stored by nodes taking part in the desk occupancy application	44
4.3	This table shows the behaviour of the Laplace distribution for various possible inputs and their respective confidence level and noise variation. In practical terms it means that for any reported noisy answer, the deduction or addition of the noise variation will provide you with a close approximation (confidence level) to where the accurate answer lies.	45
4.2	This table contrasts the results for the proposed query between the accurate readings and a noisy answer, after the application of differential privacy. It explicitly includes the amount of noise drawn from the laplacian distribution based on the random number. It also demonstrates the exponential behaviour of the Laplace distribution as the random number reaches the lower or higher extremes.	45
5.1	This table presents the classification of the characteristics of the framework with regards to data utility.	62
5.2	This table presents the classification of the characteristics of the framework with regards to point of control.	62
5.3	This table presents the classification of the characteristics of the framework with regards to targeted protection.	63
5.4	This table presents the classification of the characteristics of the framework with regards to its configurability.	63

Chapter 1

Introduction

Wireless sensor networks used to have a main role as a monitoring tool for environmental purposes and animal tracking. This spectrum of applications, however, has dramatically grown in the past few years, with solutions being deployed in health care, monitoring patients with cardiac disease for example, for building monitoring and in energy saving tools [57, 15, 78]. This growth in the number of new areas can be attributed to several factors, including the constant cost reductions and increase in processing power of the equipment required to build such systems.

These factors have shifted the previously extremely resource-constrained data collector (node) to a not so restricted part of the network, potentially allowing the execution of more complex operations, instead of the simple read-and-transmit mode of operation. Although this expansion allows constant data collection through different sensors (e.g. motion detectors, thermometers, cameras, accelerometers, GPS location), it also raises privacy concerns, with the potential of collecting data about individuals without their knowledge.

Once data is collected, it is frequently sent towards the base station of the network, where it can then be heavily processed and analysed in order to take further decisions. The base station acts as the entry point from the external world to the network. Normally, additional operations occur during the transmission phase, such as in-network processing and data aggregation. These operations aim to reduce the amount of data being transmitted, by pre-processing them or aggregating the values received from other nodes in a single value, therefore reducing the communication overhead. However, at the same time, if privacy is not addressed, each entity responsible for relaying the data needs to be considered as trusted.

End-to-end encryption between each node and the base station could eliminate the risk of intermediate nodes breaching others' privacy. It still, however, does not eliminate the problem. When the data reaches the base station and gets stored, again

a high degree of trust is necessary on the handling and use of this data. Several points of blind trust could be avoided if raw data could be usefully utilised, without leaving the place where they were initially collected and therefore causing privacy concerns to the individuals being sensed.

So far, very little research has been done into mechanisms for achieving control over the propagation and use of sensed data. The main goal rests on how to provide a viable solution in which, despite the raw data remaining where it was collected, within the nodes, the existing sensing operations do not suffer due to new restrictions, which could affect the utility of such networks. In addition to that, it is still necessary to take into account the ever-decreasing, but still restricted, resource constraints of nodes and other parts of wireless sensor networks.

In practical terms, wireless sensor networks are engineered with varying degrees of complexity [85]. These networks can be roughly classified according to their structure, either as standalone, multi-application or federated multi-application networks.

The simplest wireless sensor networks have tended to be standalone systems running a bespoke application that defined both the constituent nodes and all other aspects of the network. In such a deployment, hardware requirements are tailored to fit the needs of the application in question, with the application exploiting all aspects of the network. This structure remains common today.

Increasingly, however, wireless sensor networks are being deployed in a multi-application structure comprising nodes running a common middleware that allows one or more applications to run on the same infrastructure. The use of middleware offers a flexible and standardised abstraction of the low-level characteristics of the hardware, allowing data collected by each node to serve a number of applications. This increases the range of uses for a given deployment, but also has the potential to raise privacy or security concerns.

The sharing model can be extended further by allowing *federation* of the infrastructure. A federated multi-application network allows different entities to run applications across the same set of nodes, sharing resources between multiple stakeholders. This provides an economic benefit, and can lead to longer-term deployments offering a range of sensing options, but also raises even greater privacy concerns for those individuals in the sensing environment [45].

To date, research in wireless sensor network privacy has focused largely on privacy-preserving data aggregation (PPDA) protocols that protect the data collected in sensor nodes against outside observers, or limited malicious network participants.

Importantly, existing protocols have focused almost solely on standalone networks, without consideration for the more complex multi-application and federated networks.

One way to approach this problem is by looking at wireless sensor networks as a distributed database where each node holds its own data and provides an interface that allows other entities of the network, mainly the base station or query manager, to run queries using the locally stored data of each node. Although on the one hand this different way of viewing WSNs enables us to use existing privacy definitions and mechanisms to tackle the problem for databases, it is actually not a straightforward move, given the intrinsic characteristics of WSNs.

One promising technique to provide privacy guarantees to statistical databases is obtained by distorting the query responses, by the addition of noise. Such techniques attempt to obfuscate the real answer. However, one needs to consider how much noise is necessary for each scenario, balancing the utility of the data. It is not the goal of such techniques to perturb the data in such a way that it becomes useless.

The use of data perturbation to obtain privacy is not new in wireless sensor networks. [42] uses random noise in a data aggregation algorithm to obfuscate the answer to queries that are sent to the aggregator. However, such noise remains confined within each cluster of nodes, which, once the aggregated value is calculated, in a two-round sequence of messages within the cluster, all noise is eliminated. Although this approach seems to preserve privacy, the collusion of some members of the cluster could result in the revelation of the exact readings of any node's data. Additionally, such a technique does not consider scenarios where queries are directed to one individual node.

The study of data perturbation as the means to achieve privacy in statistical databases has been the target of other research [54]. The state-of-the-art in terms of privacy-preserving technique for statistical databases is differential privacy [25]. Differential privacy works on the premise that nothing or very little should be learned about someone, independently of whether she participates or not in the database in question. It displaces the existing premise that access to a statistical database should not enable one to learn anything about an individual that could not be learned without access to it, which was proven to be impossible to achieve.

This thesis proposes to develop the state of the art in privacy-preserving techniques to wireless sensor networks. The main benefit of using this data perturbation technique in wireless sensor networks is the strong and formal privacy guarantees that it provides. The challenges of such a proposal lay in evaluating these techniques as

well as well proposing new ones that take into account some of the particular characteristics that sensor networks have, such as the short-lived data storage, time-series data collection, distributed as opposed to centralised organisation, dynamic rather than static entities, reduced processing capabilities, and the fact that communication heavily contributes to energy consumption.

1.1 Why Does Data Privacy Matter in Wireless Sensor Networks?

Before addressing the main thesis question — to what extent data perturbation is a feasible mechanism for achieving privacy guarantees in wireless sensor networks — it is worth considering whether there is any real need for privacy-friendly networks. Are there situations where intrusive or indifferent networks would have a significant impact? The next section considers the impact on the involved entities in a multi-application environment, followed by two sections that give examples of scenarios where wireless sensor networks could be used, and data privacy is critical.

1.1.1 Scenario Shifting

The shift in the wireless sensor network organisation adds new entities and functionalities that make interoperability more complex from the point of view of control and therefore increasing the potential for breaches of the agreed services.

In summary, the following are the main entities that suffer changes as a result of the extension of wireless sensor network to support multi-applications:

- sensed environment
- network owner
- analysts and data consumers

For the sensed environment, like individuals and their behaviour, it is important for the system to provide guarantees that the collected data is being used only for the purpose that they have been designed for. Such control is challenged by the applications' overlapping functionality, since potentially more entities will be capable of collecting data and drawing conclusions about them. A damaging impact of an unreliable environment can result in individuals deliberately disturbing the environment

in order to avoid any potential data leakage or, even worse, not take part at all in the network. In both cases, the end result compromises the benefit that such network could bring and/or the use of the data by analysts.

From a network owner's point of view, it is important the cooperation of the individuals being sensed in the environment, so providing a privacy-preserving setting to them becomes a fundamental starting point, which in turn will provide a more reliable and useful source of information for data analysts. If we consider the financial aspect, the whole hardware deployment would be in vain if no or little engagement exists with the data producers, which will cause a similar level of engagement for data consumers, given the low quality of the collected data. These considerations might have a different outcome if one considers a corporate environment because employees could be obliged to participate and be monitored to prevent misleading behaviour. However, it is possible that if provided with proper tools that are capable of achieving expected levels of satisfaction, these corporations could be persuaded to introduce privacy-preserving measures. Also, in many countries (including EU states and Brazil) organisations have a legal requirement to protect personal data.

The last of the three entities directly inclined to worry about data privacy are the analysts and data consumers. In general, higher privacy levels results in less accurate data, forcing them to adapt their means of achieving expected results or having to lower their expectations. However, if the network lacks engagement, they will not be able to achieve their expected results either. Therefore, even if indirectly, data analysts might welcome the use of privacy-preserving techniques in wireless sensor networks.

1.1.2 Smart cities

Cities around the world are always looking for ways of improving life standards of their citizens in the most cost-effective way possible. As such, they have to make decisions on which areas to prioritise and the existence of relevant data to base their decisions on becomes a critical factor. A way of mitigating such a lack of data is the installation of a wireless sensor network within the perimeters of a city. Nodes could be installed on houses, cars, shops, streets, and so on and be capable of sensing a large range of attributes in order to guarantee the network's usefulness in a vast variety of projects. Once the infrastructure is in place, there is a broad spectrum of uses for such a network, like the possibility of extending access to such a rich ecosystem to companies in order to maximise the return for the investment, but the privacy concerns, among other concerns too, that this scenario would potentially generate are

just as big. Therefore a strong and reliable way of dealing with such a comprehensive environment is fundamental.

1.1.3 Infrastructure Monitoring (Case Study)

Wireless sensor networks are often used for building monitoring purposes, with goals varying from basic temperature monitoring to people tracking. Such applications could, for example, assist in better management of energy consumption in several aspects, influence building management through occupation rates, as well as more critical activities like assisting in building evacuation in case of an emergency. A possible application for such a scenario could be a multi-application wireless sensor network that is able to monitor desk occupancy across a university. The sensor node on each desk can be a member of multiple applications, e.g. security reading group, PhD Student, libraries, etc. Queries can be directed to the network to any of the available applications. In such a scenario, some typical queries could be:

- how many DPhil students work daily in their offices?
- how many hours is spent in the office in a day/week/month?
- what is the level of engagement/integration within and across groups?
- appliance integration: should printers be switched off or enter energy saving mode given the number of people in the building at any giving moment?
- has the reading group/seminar/tea time taken place this week? One could answer such query by querying the specific group of users.

There is no doubt that such network could generate privacy concerns regarding the inappropriate use of the data collected by the network. However, if properly managed, they could bring great benefits to everyone.

As such, a subset of this scenario is the chosen case study to be used in the remaining chapters of this dissertation. The subset refers to an office occupancy application. This case study works as follows. Nodes have been installed on each desk and are capable of detecting movement, which would flag the desk as occupied. Occupancy is measured regularly every minute. The overall motivations for such a case study have already been presented above.

From the point of view of privacy, the concern is with the protection of the presence or not of a single individual in the office. For example, it should not be possible to infer that an employee normally arrives at ten o'clock to the office, which would be

considered late. Equally, it should not be possible to assert that anyone leave their desks too often during normal working hours.

1.2 Challenges

The overall goal of providing users with privacy guarantees in a wireless sensor networks comes with a set of challenges that have to be understood and overcome. Some challenges are related to the nature of the environment we are dealing with while others come as a result of the technique we employ to achieve privacy: data perturbation. In summary, these challenges can be grouped into four main categories. The categories are as follows:

- distributed architecture: in a wireless sensor network, it is the nodes that are responsible for collecting data that will be used to answer queries directed towards the network. Therefore, any devised technique to achieve privacy in such a scenario has to take its distributed nature into account, so that, ideally, the raw data remains where it is collected and there isn't an entity that has to be trusted in order to achieve privacy.
- time-series data: wireless sensor networks are commonly used to collect data over periods of time with varying intervals levels between readings. Any series of these readings are likely to reveal more about the sensed item than any single individual reading alone. This is due to the correlation that a reading has with those nearby ones. For example, in the desk occupancy scenario, the correlation could be down to individuals staying on or off their desks for longer periods of time, resulting in a long sequences of readings with unchanged values. This characteristic, like the distributed architecture discussed above, goes against the original goal of differential privacy, which was to protect static data like census data.
- utility/noise levels: differential privacy relies on data perturbation to achieve privacy, which, in practice, translates to the addition of noise. It is fundamental to not only be able to maintain acceptable levels of utility but also offer a flexible way of adjusting such levels for better accuracy or better privacy.
- limited-resource environment: the way that the challenges above are addressed have to take into account the resource-constrained nature of wireless sensor

networks. The main limitations are in the energy consumption, processing power, storage and network communication areas.

1.3 Contributions and Dissertation Structure

A shift of paradigm, as previously presented, is likely to require new supporting tools and techniques to achieve matching performance in all sorts of functionalities around the correct functioning of a wireless sensor network. With that in mind, the thesis sets out to investigate how it affects data privacy as a whole, ranging from the point of view of the user being subjected to the sensing environment to that of the network owner.

In that context, the first contribution of this dissertation is an analysis of existing techniques that have been proposed to provide privacy-preserving data aggregation. Such investigation concludes that such techniques are not suitable for the purpose, with stronger implications for multi-application and federated wireless sensor networks, indicating the necessity of some form of uncertainty in order to address the identified issue: the total reliance on data aggregation as a form of privacy guarantee.

In order to add uncertainty, we use differential privacy, the state of the art in the statistical database field, as the means to achieve privacy. This definition, however, was not originally designed to work in a distributed environment and to deal with time-series data. Therefore, the second contribution comes in the form of new approaches to bring the achievements of differential privacy into the wireless sensor network scenario.

The third and main contribution of this dissertation comes in a form of a framework to assist in the designing of privacy-preserving data aggregation protocols for wireless sensor networks. Its goal is to provide a flexible way to address the required privacy guarantees while at the same time suit the application needs. The framework combines the approaches previously proposed with key characteristics of wireless sensor networks in order to do that.

This dissertation is organised as follows. Chapter 2 summarises the literature on wireless sensor networks and describes the existing techniques that have been proposed to achieve privacy, in the form of data aggregation protocols, in such networks. It also presents the relevant background research in data privacy, which includes a thorough description of differential privacy and its literature. Chapter 3 performs an evaluation of existing privacy-preserving solutions and demonstrates that they do not

provide the expected privacy guarantees, affecting all three types of network organisation. Such a finding indicates that the use of some form of uncertainty is required in order to provide some level of privacy. Chapter 4 analyses the effect that time-series data has in differential privacy and proposes two contrasting solutions that, when combined, can offer a very flexible balance between utility and noise levels. This chapter also addresses the distribution architecture challenge, with the solution relying on the central limit theorem. Chapter 5 combines the proposed techniques and presents them in a form of a framework to assist in the designing of privacy-preserving data aggregation protocols to fit the characteristics of the target network and fulfil the application's needs. This framework is then evaluated in Chapter 6 by tackling real-world applications. This dissertation concludes by presenting a summary of the contributions and suggestions for future work in Chapter 7.

Chapter 2

Wireless Sensor Network

The first known use for wireless sensor networks began in late 1968, in the military operations for the Vietnam War [15]. The entire technological solution is known as “Igloo White”. In the occasion, aeroplanes would fly over the enemy territory and drop a few hundreds of nodes there. Initially, they had the objective of revealing any enemy movement, but as the enemy’s areas were invaded, the small devices would assist in the communication and tracking of troops. Therefore, wireless sensor networks would provide, at different stages of the battle, valuable insight information about the enemy’s location; later working as a communication infrastructure. This mode of operation went on for years and it changed the way the dynamics of battlefields worked.

As years went by, new scenarios for deployment of wireless sensor networks emerged and that provoked the start of research in academia. One of the first real-world deployments of a wireless sensor network for academic purpose occurred in 2002 [57]. The goal of the project was to study the environmental conditions of a particular species of bird called Leach’s Storm-petrel living in Great Duck Island. From there on, innumerable other projects have been carried out, posing several research challenges, including privacy concerns.

This chapter presents a literature review on relevant topics of the wireless sensor networks field. It starts by describing the basic architecture and characteristics that of a wireless sensor networks (section 2.1). Next, the organisational structure of wireless sensor networks is covered (section 2.2). Subsequently, a glance of middleware and its relevance is presented (section 2.3). It then describes the function of data aggregation in such networks and its strong relation to privacy-preserving mechanisms (section 2.4). Finally, the chapter covers privacy-preserving research in the context of wireless sensor networks (section 2.5).

2.1 Basic Architecture and Characteristics

A basic architecture of a wireless sensor network contains a sink and several nodes. The sink acts as the entry and exit points for the communication of, respectively, queries and their responses to the external world. Nodes are responsible for collecting data that will be transferred across the network until it reaches the sink and then presented to entities outside the boundaries of the network.

Nodes are normally low-cost devices with limited processing capabilities. It is invariable to build tamper-proof nodes mainly due to costs of manufacturing. They are normally deployed in larger quantities and are most of times taken as disposable equipment.

They normally run on batteries and therefore have a limited energy source. This characteristic adds to the fact that nodes have low processing power and therefore offer a constrained running environment. In an ordinary deployment scenario, communication is the operation that consumes the highest amount of energy [68].

Wireless sensor networks are distributed autonomous systems with a strong capability of self-organising. This characteristic permits the deployment of networks organised in a static or dynamic manner. The density of nodes varies according to the application and objects being sensed. Packet loss is a common side effect in such a distributed architecture. This also means that the communication path can cross several nodes before any information reaches the sink.

The points highlighted here superficially cover relevant characteristics for the present research. For a more thorough coverage of basic aspects of wireless sensor networks, there are several surveys available, such as [3], [6] and [84].

2.2 Organisational Structure

Wireless sensor networks are grouped into three main categories in terms of their organisation [85]. They vary from standalone single-purposed networks to federated multi-application ones. Next, a more detailed view of the three categories are presented.

2.2.1 Standalone Networks

This is the first and simplest form of wireless sensor networks. They are characterised by running specialised standalone systems tailored for addressing the requirements of

a single application and/or use case.

In such deployments, the main goal is to fulfil the needs of the application in question. This single application is responsible for exploiting all aspects of the network. Despite constituting the characteristics of initial deployments of wireless sensor networks, they remain common today.

2.2.2 Multi-Applications Networks

Increasingly, wireless sensor networks are being deployed in a multi-application structure comprising nodes running a common middleware that allows one or more applications to run on the same infrastructure.

The use of middleware, as presented in section 2.3, offers a flexible and standardized abstraction of the low-level characteristics of the hardware, allowing data collected by each node to serve a number of applications. This increases the range of uses for a given deployment, but also has the potential to raise privacy or security concerns.

2.2.3 Federated Multi-Application Networks

The sharing model can be extended further by allowing *federation* of the infrastructure. A federated multi-application network allows different entities to run applications across the same set of nodes, sharing resources between multiple stakeholders. This provides an economic benefit, and can lead to longer-term deployments offering a range of sensing options, but also raises even greater privacy concerns for those individuals in the sensing environment [45].

2.3 Middleware

As the deployment of wireless sensor networks increased, there was a necessity to make this process easier and less time consuming. That is when the concept of middleware for wireless sensor networks started appearing. The use of middleware facilitates the deployment of applications by offering a range of components to deal with low-level characteristics. Wang et al. [81] classified these components in four categories: programming abstraction, system services, runtime support and QoS mechanism.

From the application's point of view, the programming abstraction is the most important component. This abstraction is what allows an application to interact with

the node itself and other parts of the network. The other components generalise the mechanisms required to support the correct functioning of nodes and allow them to interact with other nodes and the sink.

There are plenty of solutions for middleware. A common way to differentiate them is based on the way they handle the data they generate. The following are the most common approaches with examples: database (view the whole network as a database and uses SQL-like languages to query the network) [74], event-based (uses an asynchronous communication model) [75], application driven (reduces the middleware's involvement in the network layer by giving greater control to the application) [44], modular (an application is composed of several smaller modules and therefore facilitates software updates) [55], virtual machine (provide a smaller set of instructions for applications) [53] and tuple space (similar to the database approach but with a different query paradigm) [16].

Middleware is an area particularly interesting to consider when investigating ways to provide privacy to applications in wireless sensor networks. In the same way that happens to network communication and others aspects of the network, privacy could be tackled in a more generalised way by being integrated as part of the middleware as a service.

Privacy as a middleware service becomes essential in networks with nodes capable of running more than one application at a time. This task becomes even more challenging if these applications are being managed by different stakeholders. Huygens et al. [46] identified a set of extra requirements required to run a shared and federated wireless sensor network. They also provide, in the form of a middleware, a high-level overview of how these requirements could be addressed. In such a scenario, being able to provide privacy guarantees becomes a challenging question that requires strong data-level mechanisms implemented in the middleware that runs within nodes.

More details regarding the overall characteristics of middleware in the context of wireless sensor networks can be found in [81] and [64]. For surveys on adopted approaches and current challenges refer to [41] and [59]. Finally, a thorough comparison of middleware solutions can be found in [58].

2.4 Data Aggregation

The idea of aggregating the data being transmitted from nodes towards the sink attempts to improve in several fronts in wireless sensor networks. Firstly, the network

around the sink, depending on topology and size of the network, could easily get congested if several nodes near the extreme points of the network attempt to transmit their stream of data at once. Secondly, it has been demonstrated that network communication is much more expensive than local processing in energy consumption terms [68], so the reduction in the number of transmissions helps in building better energy-aware networks. Thirdly, it is possible to adopt mechanisms to eliminate redundancy and therefore also improve energy efficiency.

Fasolo et al. [10] present a survey covering in-network aggregation techniques. They define the in-network aggregation process as follows: *In-network aggregation is the global process of gathering and routing information through a multihop network, processing data at intermediate nodes with the objective of reducing resource consumption (in particular energy), thereby increasing network lifetime.* The survey also presents suitable classification criteria and evaluates eight well-known protocols using this criteria.

An important aspect to consider when using data aggregation is the impact it has in the overall behaviour of the network. Krishnamachari et al. [50] analyse the reduction in energy consumption and introduction of delay tradeoffs that the data aggregation process involves and how they are impacted by several factors of a network.

There has been plenty of work on securing data aggregation operations from external entities as well as compromised nodes. It includes several protocols [11, 4, 22, 83] and surveys [5].

As an example, Albath et al. [4] propose an aggregation protocol that achieves confidentiality, integrity and availability for in-network aggregation in wireless sensor networks. It uses homomorphic encryption and additive digital signatures to achieve that. The missing part is a mechanism that, while achieving the previously-stated goals, also provides privacy guarantees for the objects being subjected to the sensing environment. Using this model, it is possible to prevent the leakage of information along the path towards the sink, but no control, however, can be assumed over the data as it reaches its end point.

Another relevant area of data aggregation research is the one that considers the current WSN's infrastructure to determine the best approach to collect and transmit the data. It also takes into account the amount of error the user can tolerate. In this area, some of the relevant research contributions are: [9] and [80], which investigate the trade-off between energy and accuracy and [71], which studies the trade-off

between delay and energy for data gathering and aggregation in wireless sensor networks.

2.5 Privacy in WSN

Privacy has been investigated at the contextual as well as at the data level in wireless sensor networks. The former has received much more attention from the research community when compared to the latter.

An interesting common characteristic between the two is that in both cases there are solutions that make use of a data aggregation mechanism to achieve some level of privacy.

2.5.1 Contextual Privacy

Contextual privacy in wireless sensor networks attempts to protect the relations between communicating parties from observations.

Existing mechanisms attempt to prevent the breach of the location of valuable item that is being tracked by a wireless sensor network [47], protect against behaviour analysis of the area being sensed [77], hide the location of the base station [1] or other elements of the network [63], among others.

In an attempt to protect the location of the sink, Conner et al. [14] present a protocol that uses data aggregation and a decoy sink. In normal operation, nodes would send the data to the elected decoy sink for it to be aggregated first and then the decoy sink forwards the aggregated value to the real sink. The goal of the protocols is to reduce the amount of traffic near the real sink and therefore make traffic analysis more difficult for an attacker. This protocol shows the use of data aggregation in contextual privacy.

2.5.2 Data Privacy

Data privacy in wireless sensor networks has been achieved mainly through the use of data aggregation. However, one characteristic that appears on most protocols is that they either use an attack model that considers the sink as a trusted entity or expect to maintain 100% accuracy. Chapter 3 will demonstrate how 100% accuracy can actually cause harm to the objects being sensed in terms of their privacy.

According to Bista et al. [7], a privacy-preserving data aggregation protocol designed for wireless sensor network must address the following characteristics to be acceptable in the wireless sensor network domain:

- eavesdropping and privacy preservation: the former deals with external entities trying to eavesdrop data from nodes while the latter also ensures data privacy against trusted nodes
- data pollution and data integrity: to be able to detect undesirable data modification during the data aggregation steps as the information travels along the network
- efficiency: bandwidth and energy efficiency
- accuracy: be able to deliver accurate answers even with high packet loss
- QoS support: fundamental requirement for most applications; it covers things like tolerance to delays
- idle time: how nodes at different parts of the network operate in relation to their active/inactive activities
- dynamism: support for dynamic topologies
- aggregation functions: number of aggregation functions a protocol supports

As can be seen from the above, accuracy appears as a must-have in a privacy-preserving data aggregation protocol. However, as will be demonstrated in Chapter 3, this characteristic, if not correctly applied, ends up compromising the privacy of those objects being sensed by a node.

Existing privacy-preserving protocols can be classified according to the following categories [7]:

1. perturbation-based protocols: make use of data perturbation in order to guarantee data privacy. Protocols classified under this category will be presented more in-depth below.
2. data shuffling: are characterised by splitting the data into a several parts and transmitting each part through different path towards the sink. The number of parts for the data acts a configurable privacy level. SMART is an example of such a type [42]. There is an improved version of this protocol called iPDA [43]. iPDA offers an integrity control on top of the guarantees provided by SMART.

3. homomorphic encryption: nodes encrypt their data before transmitting to their parents. The property of homomorphic encryption allows intermediate nodes to process aggregation operations over encrypted data.

Existing perturbation-based protocols use only temporary perturbation for protecting the data from intermediate nodes. The perturbation is applied in such a way that when the data reaches the sink, none of the existing perturbation is still considered towards the final aggregated result. Excluding the protocols of the PHA family [86], all other protocols claim to protect against outsiders as well as insiders.

An example of this claim can be found in the CPDA protocol [42]. CPDA creates clusters closer to outer nodes of the network and runs a solution for the *average salary problem* [73] within each cluster. Each node applies a temporary data perturbation for its reading and transmits it during the first round of the protocol. After the second round, each node removes the perturbation initially applied. This process occurs in such a way that no one within the cluster as along the communication path is able to discover each other's data. The cluster head can then transmit the aggregated result to its parent.

2.6 Privacy-Preserving Data Aggregation Protocols in WSN

This section presents the goals and main approaches adopted in wireless sensor network to address the privacy concerns.

2.6.1 Goals

Privacy-preserving protocols in wireless sensor networks aim to preserve the privacy of individual nodes against some combination of the *sink*, the node that aggregates values reported by other nodes, and against other nodes in the network. Different approaches have tended to focus on some combination of these, with mixed results.

The protocols shown here make various trade-offs between communication complexity, computational requirements, integrity of data, and security.

2.6.2 Clustering

Privacy-preserving clustering, illustrated in Figure 2.1, functions by forming disjoint subsets of nodes, each of which calculates an aggregate sum of their data before it is

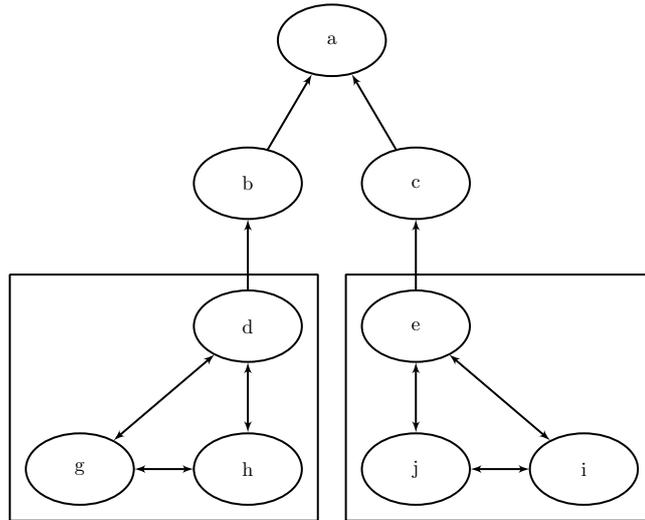


Figure 2.1: Private clustering in WSNs

sent to the sink. A variety of approaches are possible to achieve this aggregation, but a popular approach [42] makes use of a variation of the *average salary problem*. This algorithm, a simple instance of the more general secure function evaluation problem, allows nodes to sum their individual values without leaking any more information than the aggregate itself. The desired effect is that neither the sink, nor any node in the cluster, can learn the exact value of any individual node unless $n - 1$ nodes in a cluster of size n collude.

An advantage of the clustering approach is that it prevents both the sink and any individual nodes in the network from learning any single node’s values, at the expense of the bandwidth required to form clusters and perform the secure data aggregation.

2.6.3 Slicing

Slicing, introduced in [42] and then expanded in [43], chiefly aims to prevent individual nodes in the network from learning the values reported by any other nodes.

To achieve this, a node divides its values into a number of randomly-sized slices and selects multiple paths through the network, as illustrated in Figure 2.2. Each slice is sent via a different path, and added to the total sum calculated by each intermediate node, which acts as an aggregator until the value reaches the sink. The number of paths that each node sends its data acts as a configurable parameter to the required privacy level. This simple mechanism aims at providing confidentiality against other nodes in the network as well as the sink.

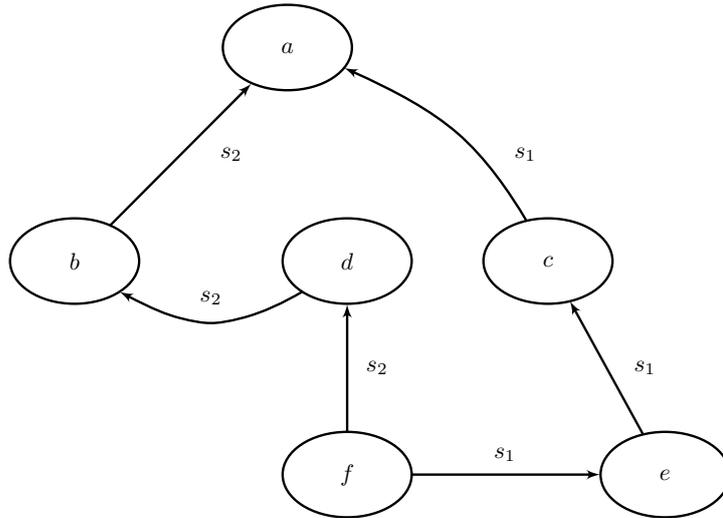


Figure 2.2: Private slicing showing the path of two private data ‘slices’ travelling across the network from f to a .

2.6.4 Privacy Homomorphisms

Privacy homomorphism uses the well-known homomorphic properties of certain public-key encryption systems to aggregate data in transit without revealing individual values. Again, this mechanism provides protection against external attackers and malicious nodes in the network, but does not prevent the sink from learning individual values.

Homomorphic encryption schemes allow manipulation of message plaintexts via the corresponding encrypted ciphertexts, enabling operations such as aggregation, or summation, of messages to be performed without decryption. Many well-known encryption schemes allow restricted homomorphic operations; in the Paillier scheme [35], for example, the multiplication of two ciphertexts under the same public key will decrypt to the summation of corresponding plaintexts, whilst raising one ciphertext to the power of another will decrypt to the product of the plaintexts.

Gentry [37] presented a fully homomorphic encryption scheme, allowing for arbitrary operations to be performed on ciphertexts. Whilst the original scheme was extremely computationally expensive, several improved schemes have already been suggested. In practice, however, even restricted homomorphism provides powerful and practical tool for privacy-preserving protocols.

In a wireless sensor network, therefore, nodes simply encrypt their values to the public key of the sink. As the message is relayed through the network, nodes can aggregate the value of any received messages simply by aggregating the ciphertexts,

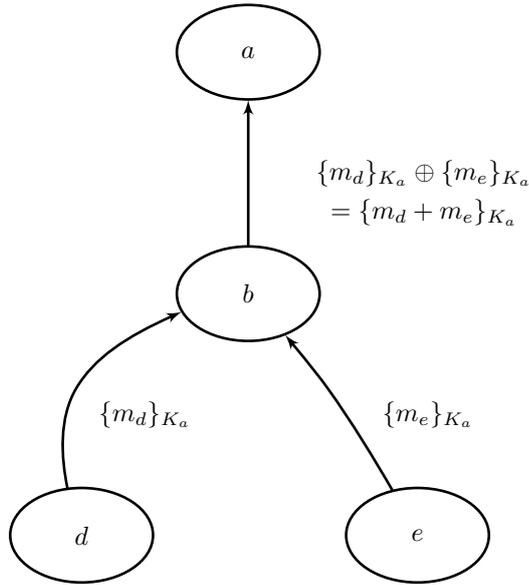


Figure 2.3: Homomorphic encryption in WSNs. Values are aggregated in encrypted form at each node.

as demonstrated in Figure 2.3. Crucially, this protects the values of any individual message from being learnt by any party except the sink.

2.7 Data-level privacy

This section presents an overview of research on data-level privacy. The overview covers two different approaches employed to guarantee privacy protection to datasets: non-interactive and interactive approaches.

In summary, all mechanisms that try to guarantee a certain level of privacy for datasets suffer from a trade-off between data usefulness and total privacy. It is not in anyone's interest to release data that do not represent a real scenario and nothing can be inferred from. But neither is the publication of raw data that may include sensitive information and cause harm to someone's privacy. The goal is to find the correct place to draw the line, succeeding on both fronts. This verification process tends to be subjective and non-trivial.

2.7.1 Models for Privacy Mechanisms

The privacy mechanisms are divided in two main approaches: non-interactive and interactive.

2.7.1.1 Non-interactive

In this mode of operation, the dataset being protected is sanitised before being released. The general goal is to anonymise the data in an attempt to hide attributes considered as personally identifiable information [66].

There is much research trying to figure out techniques to anonymise data sets and make safer their publication. One that is always referred to is k-anonymity [79]. It works by identifying the fields in the data set that could be used to identify individuals, called quasi-identifiers. It is said that the data set fulfils the anonymity goal if at least k rows have the same value for their quasi-identifier fields. To make k rows with the same quasi-identifiers some techniques are employed: reduction in the number of rows, exclusion of fields, compromising of precision, grouping of values into categories, among others. This concept has been further developed since it was first published, such as in [56], where it expands the concepts of k-anonymity and presents l-diversity.

Attacks in non-interactive mechanisms normally attempt to re-identify or de-anonymise datasets that were released after the data were subjected to a privacy-preserving technique. Examples of such attacks can be found in [76] and [79].

2.7.1.2 Interactive

In interactive mechanisms, datasets are never released. They are kept in possession of the data's holder. In order to analyse these datasets, queries are sent to the holder of the database, who then returns the answer. Therefore, the main task that interactive mechanisms have is to evaluate and decide which queries can be executed without compromising the data's privacy.

A thorough survey of techniques employed to create such mechanisms can be found in [2]. Some of the techniques are: query restriction approach, data perturbation and output-perturbation approach.

Alongside the mechanisms, there are also the research of attacks that attempt to circumvent the privacy guarantees provided by the mechanisms. An example of such attacks is the tracker attack [20]. The tracker attack works by trying to isolate parts of the database in an attempt to infer knowledge from isolated parts. An improved

version of the attack is presented in [21]. It is this attack that motivated the attack against privacy-preserving data aggregation protocols presented in the next chapter.

It was just recently that a new definition has been proposed and, so far, it has been described as a firm foundation for private data analysis [28]. It is called differential privacy.

2.8 Differential Privacy

Prior to differential privacy, the desirable property for privacy-preserving statistical databases stated that [17]: “*A statistical database should reveal nothing about an individual that could not be learned without access to the database.*” For several decades, that was the goal researchers worked towards. However, just recently has been proved impossible [25], largely due to the existence of auxiliary external information that can be combined with the data in the database.

The reformulated definition in use by differential privacy states that the fact your data is in the database does increase the chances of having your privacy breached, that is, it could have happened even if your data were not in the database. So rather than guaranteeing that a privacy breach will not occur, differential privacy guarantees that the privacy breach will not occur due to the data being in the database. This change in the definition effectively excludes possible effects current and future auxiliary information could have over the queries applied to the database.

In order to provide its strong privacy guarantees, differential privacy adds noise to the result of queries being executed over the database. That means that the database itself remains intact in relation to its accuracy. The Laplace distribution is one of the techniques employed to apply noise to queries’ results [25]. Formally, this data perturbation follows the following definition:

$$\Pr[\mathcal{K}(\mathcal{D}_1) \in \mathcal{S}] \leq \exp(\epsilon) \times \Pr[\mathcal{K}(\mathcal{D}_2) \in \mathcal{S}]$$

where, being D_1 and D_2 two databases that differ in only one entry, the probability of applying \mathcal{K} to both databases should be similar, given a multiplicative factor based on the exponentiation of ϵ .

In practical terms and taking into account the case study presented in section 1.1.3, this definition could be read as follows. The probability of querying an office and obtaining the number of occupied desks at a chosen timestamp is going to be

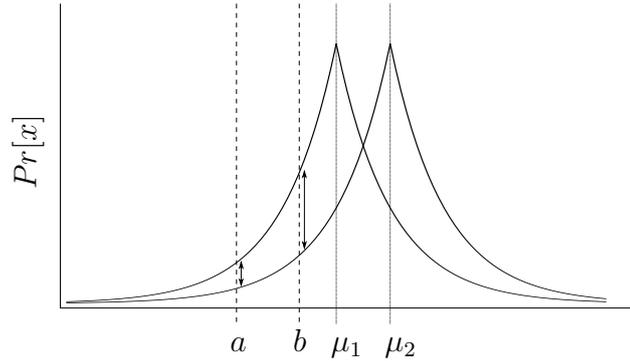


Figure 2.4: Overlapping Laplace distributions, means μ_1 and μ_2 , showing comparative probabilities of two values, a and b , drawn according to either distribution.

probabilistically similar to the result obtained should the query, for the same timestamp, exclude a desk in its calculation. The difference between these two probabilities defines the amount of privacy (budget) that is spent after each query. Figure 2.4 illustrate this example, with μ_1 representing all desks and μ_2 representing the dataset excluding one desk. The values a and b are examples of answers after the addition of noise.

Since 2006, when differential privacy was presented, plenty of research towards improving as well as attacking it has been investigated.

One example of such improvements is the use of the definition to provide privacy to distributed databases [29, 36, 69], in contrast to centralised ones. One characteristic that applies to all existing solutions for this problem is that it only applies when databases are located one hop away from the data aggregator.

One characteristic that has been identified as a possible weakness of differential privacy is when the data being protected is time-series data. This type of data can be highly correlated and therefore leak more information than initially intended. Ways to incorporate such correlation into the definition can be found in [67], [31] and [69].

A thorough description of how differential privacy works and its design decisions can be found in the following sequence of papers: [23] [32], [8] and [25]. For a survey of results and firm foundation differential privacy has established so far can be found in [27] and [28] respectively.

2.8.1 PINQ

PINQ [60] was proposed to offer practical use of differential privacy, inheriting its characteristics as well as strong privacy guarantees.

Written in C#, this library offers a rich environment for analysts to run queries and perform their analysis. Queries are written in a SQL-like language and are extremely flexible. All queries are run over live records of the database in question. Once processed, the query's results are subjected to the application of some noise. Such noise is internally managed by the library to achieve the strong guarantees of differential privacy.

The opposite side of this approach is that, despite having access to raw/unmodified data, PINQ defines a strict boundary between analysts and data.

It is important to notice here that this tool offers a unique approach for running data analysis, which eliminates the necessity of a privacy expert to evaluate the type of queries and judged what constitutes acceptable from unacceptable queries.

Additionally, PINQ can be seen as the basic foundation for comprehensive tools. Its current implementation uses previously proposed privacy techniques that are known to achieve differential privacy, like the Laplace distribution discussed above. However, such support could be extended to other techniques in order to suit other scenarios.

At the moment, PINQ works in a centralised approach, by calculating and applying noise centrally where the library is running. Such an approach is evidently not suitable for a distributed environment found in wireless sensor networks.

There was initial work carried out by McSherry et al. [61] to analyse network traces using PINQ. It was accepted that some commonly performed analysis were not straightforward to achieve, with challenges being presented to them in several occasions. However, they managed to reproduce most analyses normally necessary in such analysis, proving that the limitation of differential privacy in relation to output exactness and the fact that the analyses have to be expressed using higher level declarative language could be overcome.

Being a relatively new concept, differential privacy needs to be applied to different domains, as a way to explore its practical utility. Like PINQ, there is more work that is trying to do that, applying it to case studies from previous privacy breaches [62], botnets detection [70], among others.

2.8.2 PASTE

PASTE is a differentially private aggregation algorithm for distributed time-series data. Published by Rastogi et al. [69], this research shares two characteristics with the subject of this research: operate over time-series data and in a distributed fashion.

For addressing time-series data and avoiding high noise levels, they propose an algorithm named Fourier Perturbation Algorithm which perturbs the Discrete Fourier Transform of the answers. For addressing the lack of trusted central server, PASTE makes use of homomorphic encryption and the threshold Paillier cryptosystem [35].

Despite providing a remarkable step forward in addressing time-series data with the use of a trusted central server, PASTE does not perform well when running in a distributed manner and therefore does not suit the wireless sensor network environment. Firstly, the use of homomorphic encryption and a threshold cryptosystem make it expensive for a resource-constrained environment. Secondly, the devised Fourier Perturbation Algorithm results in intensive operations. Thirdly, it is not possible to calculate the right number of coefficients to be used in the Fourier Perturbation Algorithm. The number of coefficients has to be assumed, which could have a profound effect in the resulting noisy answers.

2.9 Summary

In this chapter we have presented an overview of areas of wireless sensor networks deemed relevant for providing data privacy guarantees in wireless sensor networks. As part of this overview, we have described the techniques used in protocols to address data-level privacy concerns. Additionally, the chapter has covered the definition of differential privacy and related topics that will be relevant to the upcoming chapters.

In the next chapter, we will demonstrate how unsuitable the techniques presented here are for addressing privacy concerns in the context of multi-application wireless sensor networks.

Chapter 3

Set Difference Attack

Wireless sensor networks are increasingly being deployed to monitor a variety of real-world environments and processes. Initially designed for military applications such as battlefield monitoring or perimeter security, wireless sensor networks are now being used to monitor industrial processes, environmental pollution, marine- and land-based ecosystems, and stock control, as well as many other purposes.

The data gathered by wireless sensor networks can in many cases be sensitive, either when considered in isolation or when combined with other data. Where individuals and their actions are monitored by a wireless sensor network we desire, or may even be legally required [34], to ensure adequate protection measures for personally sensitive data. Even when data is not directly sensitive, it is good privacy and security hygiene to prevent unnecessary dissemination of readings from individual sensor nodes.

In practice, as presented in section 2.2, wireless sensor networks occur with varying degrees of complexity. These networks can be roughly classified according to their structure, either as standalone, multi-application or federated multi-application networks.

To date, research in wireless sensor network privacy has focused largely on privacy-preserving data aggregation (PPDA) protocols that protect the data collected in sensor nodes against outside observers, or limited malicious network participants. Importantly, existing protocols have focused almost solely on standalone networks, without much consideration for the more complex multi-application and federated networks.

In summary, we are chiefly concerned with protecting, or conversely learning, individual readings from nodes in a wireless sensor network. Specifically, we are concerned with the potential to derive individual sensor node readings in a range

of network structures, but we focus on networks that support multiple applications, even in the presence of existing privacy-preserving protocols.

The remainder of the chapter is structured as follows. Firstly, we define the model and underlying assumptions (section 3.1), and introduce the notion of the *set difference attack* (section 3.2). We then explore how the capabilities and goals of existing privacy-preserving data aggregation protocols, presented in detail in section 2.6, fail to protect against these attacks (sections 3.3 and 3.4), and analyse the potential for these attacks to function in practical deployments (section 3.5). Finally, we propose an initial approach towards mitigating these attacks, and explore its implications for data collection in sensor networks (section 3.6).

3.1 System and Attacker Model

We are concerned with wireless sensor networks in which multiple stakeholders deploy applications that aggregate information provided by nodes in the network.

More formally, we consider a wireless sensor network \mathcal{W} as being comprised of a set of discrete sensor nodes $\mathcal{S} = \{s_1, s_2, \dots, s_n\}$ along with a function mapping nodes to their reported readings modelled as simple natural numbers: $\mathcal{V} : \mathcal{S} \rightarrow \mathcal{N}$. Users query some subset of sensor nodes \mathcal{A} , corresponding to those running some application, and receive a simple addition of the individual sensor (a) values: $\mathcal{V}(a) \mid \mathcal{A} \subseteq \mathcal{S}$; we further assume that both the set of nodes comprising a given application and the aggregate results of any queries are known¹.

Our goal is to protect or, adversarially, to learn the reading of any individual sensor: $\mathcal{V}(\{s\}) \mid s \in \mathcal{S}$.

We assume that applications aggregate a known subset of \mathcal{S} , reporting only an aggregate value. Intrinsically, we assume some lower limit on the size of the set $\mathcal{A} \subseteq \mathcal{S}$ in order to prevent trivially requesting the value of an individual node. We will show later how this simple defence is ineffective.

We consider two attacker models based on the standard *global passive attacker* commonly used in the field of privacy-enhancing technology research. This attacker is able to observe, but not decrypt, traffic passing between nodes but cannot alter,

¹While this may seem to place a great deal of information in the hands of potential attackers, it is a reasonable representation of existing wireless sensor network platforms. It should also be noted that the attacks we will describe remain feasible with greatly reduced, or more localized, information.

delay or drop communications; nor can this attacker compromise an individual node directly.²

We will focus on this first, truly passive, attacker restricted simply to observing the aggregate readings of applications, however the nature of our system model also naturally lends itself towards a *partially active* attacker that may deploy one or more applications subject to the limitations inherent in the system. We distinguish this from a truly active attacker in that this attacker may not drop or delay communications. These attackers correspond, respectively, to a non-stakeholder that queries the aggregate results of applications deployed by others in the network, and to a stakeholder with the ability to deploy their own applications on demand but who will not engage in openly malicious behaviour.

Further, for the current work we focus on a static moment and will not analyse in detail the potential effects of long-term analysis of sensed values. We will, however, make some mention of the effects of timing with respect to node availability in subsequent sections, but leave detailed investigation of this for future work.

The model we have described here represents recent research in federated wireless sensor network design, for example the work of Leontiadis et al. [52].

3.2 The attack

A set difference attack exploits the intersections between the sets of sensors comprising applications to discover scenarios in which individual nodes, or small clusters of nodes, are isolated.

The simplest form of this attack is demonstrated in Figure 3.1. The node coverage of two small applications is delineated by the light-grey regions. The first application covers the set $\{b, d, e\}$, and the second the set $\{b, c, d, e\}$. An application querying aggregate results from these two applications can trivially subtract the aggregate of the first application from the aggregate of the second in order to learn the exact value of node c .

²Note that this attacker differs from the common Dolev-Yao attacker in security protocol literature in that it cannot affect messages in transit.

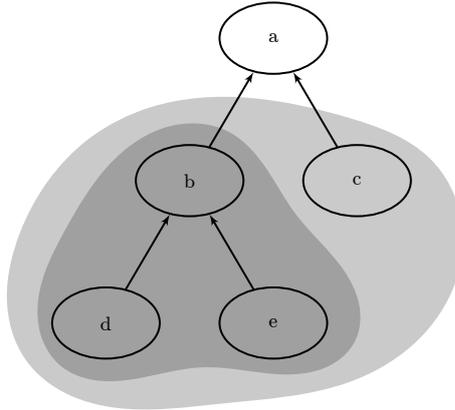


Figure 3.1: Simple set differences in a WSN.

This form of attack has some similarities to known attacks in statistical databases, known as *tracker attacks* [20], as well as to attacks against mix-based anonymous communications systems in the form of $n - 1$ attacks [38]. In Section 3.3 we will explore more complex scenarios in which these attacks apply.

An interesting feature of these attacks is that they rely only on consideration of aggregate values reported to a sink, and thus make no attempt to read data as it passes across the network. Crucially, as we will demonstrate, this makes these attacks applicable against most well-known families of privacy-preserving protocols for wireless sensor networks proposed in the literature.

Having introduced the set difference attack, we will now describe the most common approaches towards protecting privacy of individual sensor node readings in wireless sensor networks before showing how the attack applies against these protocols.

3.3 Set Difference Attacks in Detail

The set difference attack seeks to isolate nodes from aggregates in order to breach the privacy of their data. In practice, this can be achieved in one of two ways.

Firstly, the segmentation of the network caused by multiple applications running across disparate set of nodes can be exploited. An attacker therefore combines aggregate values of multiple applications in order to isolate single nodes. It is this approach on which we focus in the current work.

Secondly, an attacker can exploit the participation of nodes in aggregates taken at different moments in time. If nodes cannot be guaranteed always to report their

values, then the aggregate value of an aggregate may include or exclude certain nodes when queried at different times. This behaviour is extremely likely to result in a set difference attack, as the set of nodes being queried is likely to remain largely the same.

These two approaches can be employed in isolation, or combined by an attacker. If the attacker can learn predictable patterns of node uptimes across the network, or can observe that certain groups of nodes are more likely to be clustered in applications, the effectiveness of the attack is increased.

While the example set difference attack shown in Figure 3.1 is relatively simple, the attack itself is surprisingly powerful and hard to avoid. In addition to the simple isolation of a node via finding an appropriately-sized subset, four additional cases are worthy of mention.

3.3.1 Isolated Cluster

Trivially, the set difference attack allows us to reveal the aggregate value of an isolated *cluster* rather than an individual node. While this is not a privacy risk equivalent to the leakage of an individual node value, the leaking of the aggregates of a small set of nodes may still be in violation of the privacy goals of the system.

3.3.2 Combined Subsets

Although the most basic form of set difference attack comes from observing a subset of size $n - 1$ of a given set of size n , it is of course possible for the subset to itself be the union of a number of disjoint subsets as illustrated in Figure 3.2.

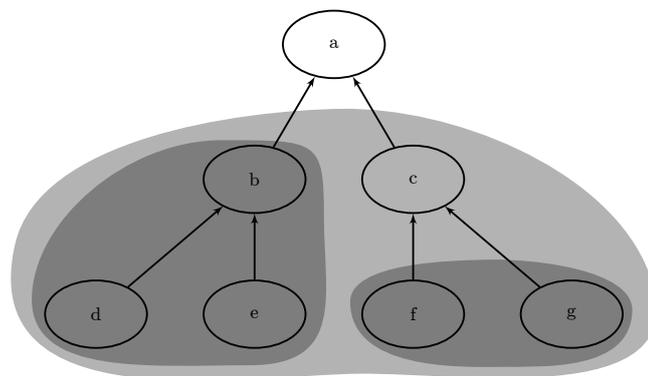


Figure 3.2: A set difference attack combining multiple disjoint subsets.

This possibility greatly increases the likelihood of observing a successful set difference attack. Observed aggregates can be stored by an attacker and combined whenever new appropriate aggregates are found. Of course, this application of the attack is highly time-dependent.

3.3.3 Total Set Coverage

In general, set difference attacks are not possible where observed subsets overlap, as this includes multiple unknown values in the combined aggregates. It is possible, however, to calculate values through gathering complete collections of sets that intersect on all but one of their elements. By gathering every possible subset of size $n - 1$ from a set of size n , we can derive *all* individual values that comprise the set. The aggregate values reported for each subset form a simple system of simultaneous equations that can be solved for each individual value.

The difficulty of performing this attack relies on the size of the subsets that we observe, as we require all $\binom{n}{n-1}$ subsets of the observed subset of size n . While we will not perform a detailed analysis of the likelihood of this scenario, it relates to the well-known *coupon collector's problem* [33] in which a collector seeks to obtain a complete collection of a set of coupons, one of which is randomly included with each purchase of a given product. It is known that the number of purchases required before obtaining the entire set of coupons is of the order $n \log(n)$, where n is the number of coupons in the set. For large networks, this scenario quickly becomes highly unlikely, however it may be practical in smaller networks or those networks where applications are likely to sample from small sets of related nodes.

3.3.4 Attack Recursion

The result of a successful set difference attack provides information to an attacker that can lead to further successful attacks. By learning the value of an individual node, or of a small subset of nodes, an attacker can remove that node's value from any observed aggregates in the network. This may itself reveal further isolated subsets that can themselves compromise further sets. As such, the attacker can potentially 'recurse' through several further attacks once any one attack has succeeded.

3.4 Attacking Existing Protocols

Existing approaches to protecting privacy in wireless sensor networks focus, to varying degrees, on manipulating data as it flows from a sensor to a sink. Clustering approaches aggregate data by combining values that are then forwarded in aggregate form. Slicing approaches split data unpredictably and randomly re-route individual portions along different paths. A privacy homomorphism encrypts data in such a way that it can be unobservably aggregated in transit. The set difference attack, however, is entirely agnostic with respect to the flow of data; instead it operates purely through examination of the final aggregate, undermining the assumptions of existing protocols and therefore rendering them vulnerable.

Clustering, in particular, may actively aid in the application of a set difference attack. As presented in [42], the choice of node clusters is random. A result of this is that multiple requests by an application are likely to result in the selection of different clusters. These, in turn, can directly cause the isolation of nodes in precisely the way envisioned in our original statement of the attack.

Slicing approaches and solutions based on homomorphic encryption share similar patterns of failure. The values of each node are protected, or at least obscured, whilst in transit, however the results are still accurately aggregated by the application. Whilst the existing protocols do provide some measure of protection against the specific threat model of an adversary that seeks to learn values in transit, they are ineffective against the attacker described in Section 3.1.

Ultimately, it is the requirement for accurate data reporting that results in the success of the set difference attack, and it is therefore this feature of the network that must be addressed by protocols in order to prevent the attack.

3.4.1 Node Availability

As we have mentioned, it is possible to perform a set difference attack through node availability rather than overlapping applications. In this case, an application that has a known, fixed set of nodes, but for which certain nodes are not always available, the absence or presence of individual nodes can clearly lead to similar attacks. Most notably, this attack will be effective even in single-application networks.

The inclusion of a time dimension in the attack clearly adds a layer of sophistication to the attack. If the availability of certain nodes is predictable, queries can be specifically targeted to take advantage of this data. Interestingly, an individual node

has little power to prevent this attack in the general case, as it will be offline when the attack effectively occurs.

A slightly more nuanced version of this attack, which we leave for future work, comes from the predictability of individual nodes over time. Clearly, certain types of sensor readings will vary predictably with time, such as light levels during the day. This can lead to predictable patterns of data being reported for each node. A more sophisticated variant of the attack would be to infer variations between nodes due to the predictable variations in aggregate reports. Similar concepts have been suggested in the context of tracking of users in online anonymization services [65], however we will not consider this potential further in the current work.

3.5 Feasibility of Set Difference Attacks

To investigate the feasibility of the set difference attack in practice, we adopt a simulation-based approach, employing abstract networks of varying size based on the system model of Section 3.1.

For each experiment, randomly-sized subsets of the network were repeatedly drawn at random. Each subset was stored and compared against all previously-drawn sets, individually and in additive and subtractive operations, to determine if a set difference attack had become possible. An attack was considered to have occurred as soon as any individual node could be isolated due to the combination of any number of previously-drawn sets. Sets were drawn continually until the attack succeeded, whereupon the number of sets drawn was recorded. To prevent trivial attacks, subsets were restricted to being of cardinality three or greater, up to the size of the network. To ensure a sufficiently low error margin for the mean, experiments were repeated in the order of one thousand times for each network size.

As a practical example of a successful simulated attack, consider a network of five nodes, $\mathcal{S} = \{a, b, c, d, e\}$, in which each node is equally likely to be selected. During a particular simulation run, three subsets were drawn: $\mathcal{A}_1 = \{a, c, e\}$, $\mathcal{A}_2 = \{a, b, c, d, e\}$ and $\mathcal{A}_3 = \{a, b, d\}$. The isolation of a node occurs by subtracting the aggregate of \mathcal{A}_2 from that of \mathcal{A}_1 , which is then summed with the aggregate result of \mathcal{A}_3 . This sequence of operations will result in isolating the reported reading of node a . Note that both operations, additive and subtractive, take place over the aggregate result of a query sent towards a subset of nodes, and not as subset operations.

The results of the simulation, showing the mean number of sets drawn before a successful attack, are presented in Figure 3.3.

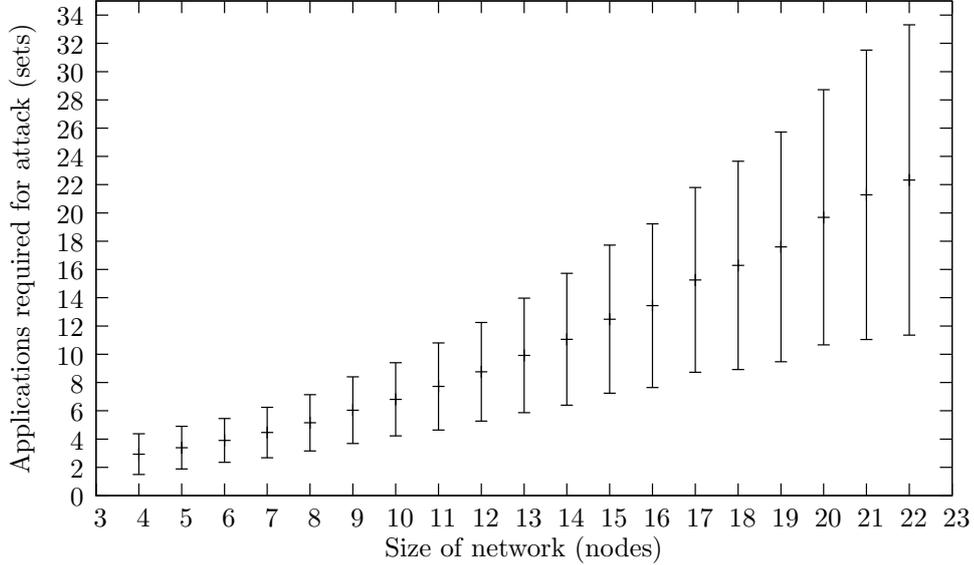


Figure 3.3: Mean average and sample standard deviation of randomly-chosen sets required in networks of varying size before a successful set difference attack.

As can be seen, the mean number of subsets required before isolating a single node is relatively low in the simulated networks, typically being lower than the number of nodes. The growth of the function does, however, appear to be more than linear, as might be expected due to the rate of increase of possible subsets. While this suggests that extremely large networks may not be easy targets for the set difference attack, networks of the size commonly seen in practice may well be vulnerable. Despite this it is worth noting that the lower bound for the required number of sets remains two, and simulation demonstrated such attacks occurring in practice for each network size that was tested.

Calculating the appropriate sets required to conduct an attack is itself extremely computationally expensive. As each new set is drawn, it must be combined with all existing sets, both in an additive and subtractive sense, to determine if an attack has been successful. The stored sets, and the number of comparisons required, grow exponentially. There are various optimizations to reduce the number of sets that must be stored and compared, and various ways to exclude sets that cannot take part in a successful attack, however the underlying complexity of the problem cannot be avoided.

For the sake of practicality, it will be possible to take a heuristic approach towards discovering set overlaps that, despite missing a proportion of successful attacks, will still result in isolating individual nodes. It is also the case that, as we have discussed, real-world networks present time constraints on the freshness and availability of sensor readings. This will present challenges to the attacker in discovering appropriate sets during a given time window, but will also greatly reduce the complexity required to perform the attack.

3.6 Preventing Set Difference Attacks

As has been demonstrated, existing protocols cannot protect node-level privacy against the set difference attack under reasonable assumptions. This is largely due to their reliance purely on data aggregation to provide privacy guarantees at the node level. In this section, we will consider the use of *data perturbation* to provide effective privacy guarantees, and examine the accuracy tradeoff that these approaches cause.

3.6.1 A Note on Fixed Clustering

Before we discuss data perturbation it is worth first mentioning one potential avenue of protection against set difference attacks, and explaining why this approach is unlikely to be of great use.

One approach that initially seems attractive for protecting against this form of attack is to enforce fixed-size clusters, or fixed size applications, and ensure the subsets of nodes resulting from these are either entirely disjoint or entirely equal. By doing so, individual nodes cannot be isolated, and thus the attack fails.

There are two major problems with this approach. Firstly, it places unreasonable constraints on applications in a multi-application or federated network. Specific deployments are likely to require specific node coverage, and the inability to choose other than a given fixed topology for applications could seriously hinder the flexibility of the network.

More seriously, this approach still cannot protect against attacks due to unavailable nodes. As is mentioned in Section 3.4.1, set difference attacks can arise from both predictable patterns of node availability, and potentially from predictable patterns of sensor readings. Neither of these factors will be affected by fixed-size clustering, and

thus cannot provide full protection against the attack. We will therefore focus on other, fundamentally different, approaches.

3.6.2 Data Perturbation

To protect against a set difference attack, we propose applying random noise to sensor readings. The purpose of this is to prevent the individual value reported by a node from being meaningful even if it can be isolated by the attack. Clearly, for some applications, this *data perturbation* approach can cause an unacceptable level of inaccuracy in aggregate results. In such cases, the risks of attack must be weighed against the requirement for accurate data.

Sensor nodes can effectively obscure their data by adding random noise drawn from an appropriately-scaled symmetric probability distribution with mean 0 to their reported readings. To protect readings effectively, the standard deviation of the distribution in question should be chosen according to the possible range of values for the given reading type. Due to Chebyshev’s inequality, this ensures that the value reported by a node, including noise, effectively covers a range of values that could be reported by the node with high probability. As seen in section 2.8, there is a well-known method for selecting privacy-preserving noise optimally according to the *differential privacy* guarantee of Dwork [26], where we also discuss in more detail the notion of data perturbation.

Usefully, combining multiple readings and their associated random noise causes the aggregate noise to converge rapidly towards zero as the number of nodes increases, due to the weak law of large numbers. The aggregate therefore tends towards greater accuracy as the number of nodes in a given application increases, making the data perturbation approach increasingly applicable as the network scales.

For noise drawn from a Gaussian distribution the sample mean, representing the aggregate noise reported from each sensor, is a good estimator of the true mean. The mean standard error of the sample mean, therefore, describes the expected inaccuracy incurred by this method of privacy-preserving data perturbation. To summarize:

For the sample mean:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i \tag{3.1}$$

The mean standard error can be described as:

$$MSE(\bar{X}) = E((\bar{X} - \mu)^2) = \frac{\sigma^2}{n} \tag{3.2}$$

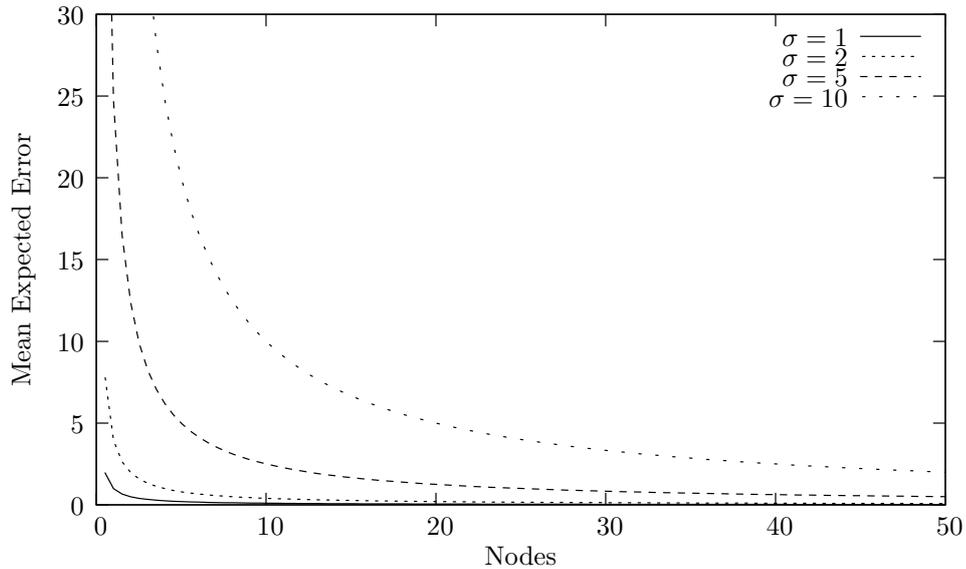


Figure 3.4: Mean standard error (MSE) for various values of σ as application size increases.

As can be seen from Figure 3.4, the expected error rapidly becomes small as the number of nodes in an application increases, even for relatively large values of σ .

3.6.2.1 Perturbation alongside other mechanisms

It is important to note that the perturbation of data in the sense we have described above is largely orthogonal to the mechanisms surveyed in Section 2.6. As such it is entirely possible, and may indeed be advisable, for nodes to cluster, slice or encrypt their data in addition to perturbing their data. In particular, this approach has the potential to improve the node-level privacy even in situations where set difference attacks are not possible, and may add privacy properties that protect against other classes of attacker.

3.7 Conclusion

This chapter demonstrated, in a form of an attack, how unsuitable existing algorithms are in achieving privacy in multi-application wireless sensor networks under reasonable assumptions. It concluded by presenting that one possibility of mitigating the attack is with the use of some form of uncertainty. Having examined an informal approach to data perturbation, the next chapter will investigate how to formally

employ data perturbation, in the form of *differential privacy*, in a wireless sensor network environment.

Chapter 4

Formally Perturbing Wireless Sensor Networks

Differential privacy, as presented in section 2.8, works by carefully adding noise to the responses of queries run over the database being protected and monitoring, through an aggregate privacy budget, how much data has been revealed from the database. Therefore, the guarantees of this definition trade accuracy in exchange for privacy and this trade-off is configurable varying according to the goals and requirements of the application. Moreover, the definition formally bounds the amount of privacy lost after each query.

Since its publication, many researchers have based their privacy solutions on what differential privacy defines as strong and formal privacy guarantees. The obvious use, which is the protection of central databases in an interactive manner, has been soon overwhelmed by numerous new applications. This range of applications varies from smart electricity metering [18] to network traffic analysis [61], including distributed computing on clusters of computers [24], and others.

In essence, differential privacy defines a single requirement for protecting the privacy of a dataset. It states that the dataset should contain one or very few rows of data related to the elementary information being protected, which could be an individual's identity for a census database or a single electricity reading in the smart metering case. Therefore, the alignment between the content of the dataset and the desirable level of protection is paramount for a valid use of this definition.

Despite this key and unique requirement, there have been cases of publications, like [60], seeking to broaden the spectrum of applications of differential privacy further, seem to ignore it, or assume a best-case scenario, and propose solutions that do not fully comply with the differential privacy definition, potentially leaking more data than they should do. It is exactly this problem that makes differential privacy

unsuitable for protecting privacy in wireless sensor networks since by design these networks capture continuous readings from the environment surrounding them.

In this chapter, we present the statistical set difference attack, which explores the granularity misalignment between data and privacy goal. The attack is first presented in its generic form and then shown in practice against some existing privacy-preserving applications, in order to demonstrate its effectiveness. Additionally, we present two ways of mitigating this attack by (1) incorporating the idea of sensitivity of the database and (2) exploring the budgeting feature in order to conform with such misalignment. We then evaluate these solution before concluding.

Notably, the attack is not on differential privacy itself, since originally the definition states, although perhaps not in the most direct form, that a protected database must contain one or very few data entries of the protected elementary level. Instead, it is directed towards solutions that make use of differential privacy in a way that is not applicable and therefore prone to privacy breaches, clearly against the goal of the protocols and their solutions. The attack in this chapter generally succeeds after executing one or two queries and certainly in no more than a couple of them, contrasting with the attack proposed by Sarathy et al. [72], which requires as many queries as the size of the dataset (see section 4.2 for an in-depth view of both attacks)

4.1 Misalignment of Granularity

One of the goals in wireless sensor networks when it comes to preserving privacy is the non disclosure of the node's reported values to other entities involved in the network as well as outsiders. Both can be equality important, mainly depending on the nature of the network and serving applications.

In line with that there is also the fact that a node generally stores time-series data regarding a single individual and therefore, by protecting the node's reported readings, one would not only be achieving the privacy goal related to wireless sensor networks described above, but also be assuring that the individual's privacy is protected. This distinction between the node's and the individual's privacy has to be addressed, with the latter generally establishing the baseline for the noise levels.

In both cases, the direct approach set by existing protocols, where noise is added to the response only based on the sensitivity of the function does not solve the problem of leaking information because there is a misalignment of granularity. Taking into account the sensitivity of the function guarantees event-level privacy whereas the

timestamp	presence
2014-03-01 12:01	1
2014-03-01 12:02	0
2014-03-01 12:03	1
...	
2014-03-01 13:59	0
2014-03-01 14:00	1

Table 4.1: This table represents a sample of the occupancy table stored by nodes taking part in the desk occupancy application

goal is either node-level, from a WSN’s point of view, or individual-level, from the individual’s point of view. It will be common for node-level and individual-level privacy to refer to the same guarantees.

4.1.1 Misalignment in Practice

To demonstrate the granularity misalignment problem, let us assume a sensor node of the case study presented in section 1.1.3, which is responsible for detecting the presence of an individual at his office desk. The way the detection takes place is irrelevant here, what really matters is that the node stores binary readings that define whether there was or there was not someone at the desk. The detection process takes place every minute and, for the purpose of this example, it stores two hours worth of readings. In summary, a node contains an *occupancy* table with two columns, timestamp and presence, which will store 120 entries after the monitoring phase is complete, exactly one entry for each timestamp within the two hours period, as shown in table 4.1. The goal is to answer the following question:

How long was the desk occupied for? or (in SQL):
`SELECT COUNT(presence) WHERE presence = 1;`

An accurate answer for this question over the suggested dataset will vary from 0, representing that there was nobody on the desk within the two hours period, to 120, showing that the desk remained occupied during the entire two hours. In contrast, if we consider a direct application of differential privacy for the same query, the standard deviation for the Laplace distribution would depend on the chosen ϵ , provided by the data analyst, and the nature of the operation being applied. The actual noisy answer then would depend on a randomly drawn value.

The random value, drawn from a truly random generator, works as the input for the amount of noise resulting from the Laplace distribution, and its value ranges from

random number	confidence level	noise variation
$-0.3 < x \leq 0.3$	60%	+/- 3.9
$-0.4 < x \leq 0.4$	80%	+/- 9.0
$-0.45 < x \leq 0.45$	90%	+/- 14.4
$-0.475 < x \leq 0.475$	95%	+/- 19.8
$-0.49 < x \leq 0.49$	98%	+/- 26.5
$-0.495 < x \leq 0.495$	99%	+/- 31.5

Table 4.3: This table shows the behaviour of the Laplace distribution for various possible inputs and their respective confidence level and noise variation. In practical terms it means that for any reported noisy answer, the deduction or addition of the noise variation will provide you with a close approximation (confidence level) to where the accurate answer lies.

-0.5 to 0.5. Being an exponential distribution, as the random number approximates the lower and higher extremes, the amount of noise increases exponentially.

With all of that in mind, let us generate a few examples of how a few answers to the query presented above looks like, assuming an $\epsilon = 0.1$:

accurate reading	Laplace noise (random number)	noisy answer
15	3.901(-0.3)	18.901
60	0.623(-0.1362)	60.623
90	- 0.310(0.0983)	90.310
115	- 12.673(0.4372)	103.673

Table 4.2: This table contrasts the results for the proposed query between the accurate readings and a noisy answer, after the application of differential privacy. It explicitly includes the amount of noise drawn from the laplacian distribution based on the random number. It also demonstrates the exponential behaviour of the Laplace distribution as the random number reaches the lower or higher extremes.

Now, if we take into account that the range to which the random number operates and that it is truly randomly generated, it is easy to define regions of confidence around the reported noisy answers and from there draw accurate enough conclusions about the individual's presence at his desk. Using the last example given above (table 4.2), one would be able to assert, after executing a single query, that the individual with the reported answer 103.673 was on his desk with 99% certainty for at least 72 minutes within the two hours period.

We are only able to create such an assertion because we can pre-calculate the noise levels for each possible random number, as it is demonstrated in table 4.3.

The content of table 4.3 plays a fundamental part in the inferring power of the statistical set difference attack. The attack is the subject of the next section.

4.2 Statistical Set Difference Attack

This attack is an extension of the set difference attack (Chapter 3). This extension equally explores the output of queries directed towards subsets of wireless sensor networks networks, but, this time, the results of the queries have to be analysed taking into account the noise levels and based on the confidence levels presented in the previous section. Node availability continues to be a valid factor for attacking a network.

What contributes even further to the effectiveness of the attack, despite the added uncertainty, and at the same time makes the two attacks similar, is that nodes would generally return the same result if the same query is sent more than once to be executed over the same dataset, so that there is no need for (1) recalculation and (2) it does not consume any additional budget. However, the attack is equally effective even if all queries are treated independently and all calculations are executed every time. That is due to the damaging effect of the misalignment of granularity of privacy guarantees.

One could argue that this attack is only possible because we are running one single query across the entire dataset at once, but that is not the case. If instead we run the query for each timestamp individually, which in practice could represent on-the-fly queries taking a current snapshot of the environment (nodes without storage), since effectively we could query the node on a 60-second interval for the current valid reading, we would indeed be able to obtain similar conclusions.

To put this into perspective, querying the network once per timestamp is a completely acceptable scenario since the queries that we run individually access different rows in the database, yet correlated, which is analogous to the data partition scenario incorporated in PINQ [60]. PINQ, a practical implementation of differential privacy, uses data partition as a way to allow multiple queries to be run as long as each query accesses disjoint parts of the dataset.

On one more note, the use of multiple queries contrasts with another attack recently published [72], that requires as many queries against the target row that they are trying to re-identify as the number of rows in the dataset. It clearly goes against

the differential privacy definition since it completely ignores the idea of budgeting and the expected data leakage after each query.

One example of an application where differential privacy has been employed with a means of making the solution privacy friendly is for network trace analysis. Data analysts with access to PING-like interfaces to raw network trace data of an internet service provider could extract information about the actual individuals behind these traces. With the use of auxiliary information, the damage is considerably worse. Access to such a dataset could be deemed differentially private, but, in reality, given the misalignment of granularity, it leaks unwanted additional information. Individual entries in the database are protected, but that is not necessarily the threat one wants to protect against.

This section shows that incorporating differential privacy in a wireless sensor networks does not address the privacy concerns raised with the set difference attack. In fact, it is worse because during the investigation of the statistical set difference attack, we were able to identify other applications that make use of differential privacy but that in reality the achieved privacy levels do not necessarily meet the expected ones. Ways to address such misalignment are the subject of the next section.

4.3 Mitigation

As it has been demonstrated, the simple application of differential privacy cannot protect node-level privacy against the statistical set difference attack under reasonable assumptions. This is largely due to their misalignment of granularity in providing privacy: node versus individual-reading levels. In this section, we present two novel solutions that address this misalignment and therefore the attack presented above: spatially-limited data grouping and enhanced budget management. These approaches provide a contrasting effects in terms of their end result (noisier and less noise respectively) but in conjunction they offer a flexible and effective manner of achieving the desirable level of privacy.

4.3.1 A Note on Query Limitation

Before we discuss the two proposed solutions it is worth first mentioning one potential avenue of protection against the misalignment of granularity, and explaining why this approach is unlikely to be of great use.

An approach that initially seems attractive for protecting against this form of issue is to enforce a query approval procedure and ensure that answers resulting from these questions do not leak undesirable information. By doing so, compromising queries cannot be executed, and thus the attack fails.

There are two major problems with this approach. Firstly, this solution would not be suitable in the context of wireless sensor networks which are autonomous and unattended systems, since it places unreasonable constraints on applications, especially in a multi-application or federated network, since queries would have to be evaluated on a case by case basis.

Finally, and more importantly, such an approach would deem it impossible to allow queries over series data due to, as it has been demonstrated in Section 4.1, the misalignment of granularity of the aimed privacy goals and the achieved privacy guarantees. Virtually any query, unless targeting a very limited set of timestamps, would have to be rejected.

4.3.2 Spatially-limited Data Grouping

This approach's goal is to add enough noise to cover up the possibility of grouping entries, therefore providing privacy guarantee at the group level. In terms of alignment of granularity, it means increasing the protection that the definition provides in order to meet the target privacy requirement. In practical terms, it means more noise, since the standard deviation of the calculation of the amount of noise is much larger.

Taking the above into account, the definition of privacy will take the following form:

$$\Pr[\mathcal{K}(\mathcal{D}_1) \in \mathcal{S}] \leq \exp(\epsilon c) \times \Pr[\mathcal{K}(\mathcal{D}_2) \in \mathcal{S}]$$

where c represents the database granularity.

This change of definition provides for a flexible solution that can be tuned to any target privacy requirement in wireless sensor networks or where the granularity of the database has to be taken into account. The selection of the right granularity levels is left for the data owners to fulfil their privacy needs. The practical consequences of higher or lower levels of granularity will be demonstrated in section 4.4.

Strictly speaking, if we were to apply the differential privacy definition in order to achieve privacy at the node rather than at the single reading levels, the granularity of the operation would have to be equal to the maximum number of readings a single node had. Therefore, to some extent, this new definition can be seen as a relaxation of

the differential privacy definition, since it will see each data grouping as its elementary unit.

The actual data grouping is quite flexible, with the decision solely depending on the data and/or application owner and the application the data will be serving. It could take the form of two hours groups or even daily grouping. The effect of increasing the grouping size is that it will generate greater noise levels. The countermeasure to that effect then becomes the combination with the next proposed approach to mitigate the misalignment: enhanced budget management.

Danezis et al. [18] achieved a similar solution for their differentially private billing solution, whereby electrical readings were grouped in order to hide their correlation and therefore increasing the granularity of the target protection for that group. This solution turned out to have a much greater application than their proposed use, as it has been demonstrated in this chapter so far.

4.3.3 Enhanced Budget Management

This approach extends the way the amount of budget is deduced from the rows of the table being queried, so that not only the queried row has its budget reduced but its neighbouring rows (in the time dimension) have their budget reduced too.

Whereas in the previous mitigation solution the flexibility was given to the data owner, for this one, it will be the data analyst who will benefit, since it is up to them to decide how to consume all the data that they potentially have access to. This is another variable, in addition to the epsilon, that data analysts have freedom to choose from that will consume the total amount of budget in offer to them.

In practical terms it means more restriction on the number of queries an analyst is allowed to run over the database. In contrast to the previous solution, here it is possible to obtain more accurate answer to queries. It will be a trade of number of queries for accuracy.

Another possibility, not discussed so far, is the possibility of linking non-consecutive time windows, like the same hour every day in a given week. Such a decision certainly depends on the nature of the data being made available, since it is especially applicable for correlated events across sparse intervals.

Assume that an analyst wants to query the current state of the network for a particular variable. The answer of that query would incur the amount of noise that is necessary to cover for the level of data grouping, which could be considerably high. However, since the analyst is only interested in that particular reading within that grouping, or maybe a range of readings but that is considerably smaller than the size

of the grouping, it is possible to compensate for those non-used entries in the form of higher accuracy and wider budget consumption. It would work as a budget-borrowing mechanism. Effectively, the analyst would be concentrating the budget that he has over the entire range of a grouping block into one or few entries. The end result is the consumption of the budget across the entire range.

4.4 Evaluation

In the last section, we proposed two solutions to address the issue of misalignment of granularity. As mentioned, the solutions have contrasting effects in terms of noise levels. When combined, they provide a flexible and practical solution for leveraging data usefulness and privacy guarantees depending on the application needs. This section aims at demonstrating these characteristics over wireless sensor network applications.

For the first evaluation, we make use of the expanded case study presented in section 4.1, that proposes an wireless sensor network with nodes installed on desks in an office building with the goal of collecting desk occupancy data. As demonstrated there, the time-series data generated by each node is intimately linked to individuals. Such data correlation between readings of each node directly exposes the effect of the misalignment of granularity in terms of privacy guarantees.

The goal of the application is to provide the average occupancy of the desks within a two-hour period. The readings take place every minute. To start with, let us consider that, after each reading, the node reports it to the network's sink, who is responsible for storing them and answering queries sent towards the network by data analysts. Figure 4.1 presents how such an application would respond to the average occupancy query. It includes the accurate answer to the query and the standard deviation of the answer based on the solution proposed in this chapter after 500 runs.

As can be seen in figure 4.1, the noisy results are fairly accurate even for considerably small networks, with higher levels of accuracy being achieved as the number of nodes grows.

For the second evaluation, we use the same scenario as the one above, but, this time, with the data fully distributed, that is, the data is collected by the nodes and it remains within its boundaries. Therefore, each node is capable of answering private queries, assuming the queries do not extrapolate the node's budget limits, in the same same way the sink could answer queries to data analysts in the previous scenario. The sink will act as a simple proxy, coordinating the communication with

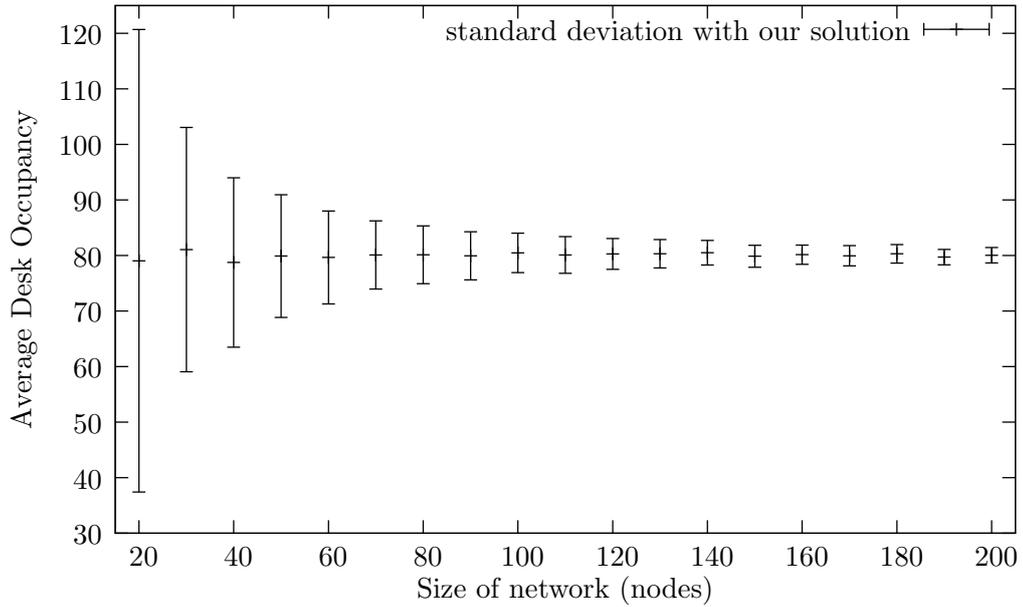


Figure 4.1: It shows the average desk occupancy over a two-hour period. The occupancy results are presented in its accurate value and the standard deviation resulting of 500 runs of the solution proposed here. These results vary accordingly to the size of the network and they are based on the data centrally located at the sink.

all nodes in order to send the query and collecting the answers sent to it by all nodes. It is then able to respond to the data analyst with the aggregated answer. Figure 4.2 presents the practical evaluation of such scenario.

Notably, the fully distributed scenario does incur considerable higher noise levels when compared to a similar network size in the centralised one, hence the bigger size of networks. However, like before, as the network size grows, the noise levels drop.

So far, we have only addressed the granularity misalignment by using the spatially-limited data grouping solution. By combining it with the second solution, however, you give back to data analysts the possibility of obtaining lower levels of noise even for considerably smaller networks as it is being shown in figure 4.3. In this example, we are only interested whether the desk was occupied eight times within the 120 minutes period: every 15 minutes. It again uses the same scenario as above.

The enhanced budget management provides for data analysts the possibility of querying a single or very few data entries rather than all entries within the size of the data grouping (sensitivity of the database). Such an approach guarantees that the effect of noise levels, in a fully distributed scenario, are only dependent on the number of nodes taking in the network, acting completely obviously to the database granularity. As it has been seen in figure 4.3, for considerably smaller networks, it

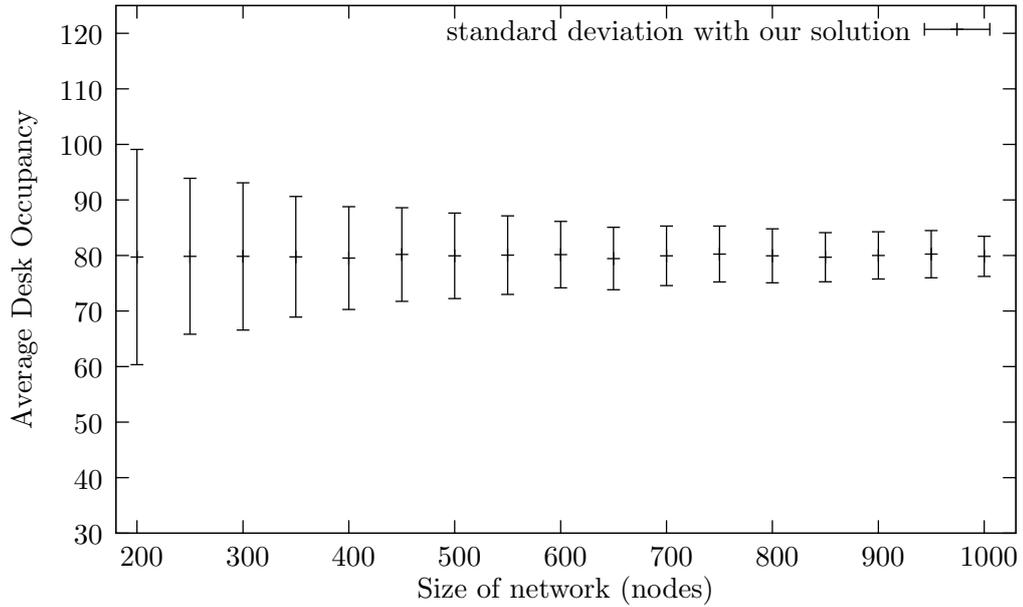


Figure 4.2: It shows the average desk occupancy over a two-hour period. The occupancy results are presented in its accurate value and the standard deviation resulting of 500 runs of the solution proposed here. These results vary accordingly to the size of the network and they are based on the data being fully distributed (only the node themselves have access to the raw data).

is possible to achieve considerable higher levels of accuracy when compare to the previous two simulation.

4.5 Conclusion

In this chapter we show that, although differential privacy has been a breakthrough in terms of statistical privacy database, it falls short for time-series data, which by itself is not an entirely new finding, but it also reveals that existing applications are using the current definition of differential privacy indiscriminately, ignoring its limitations.

Such limitation is demonstrated by the effectiveness of the statistical set difference attack, which explores the granularity misalignment between datasets and aimed levels of privacy.

In order to mitigate the attack, we propose two approaches. The first introduces the concept of database sensitivity, which extends the privacy protection to wider ranges of database entries. The sensitivity level is configurable and depends directly

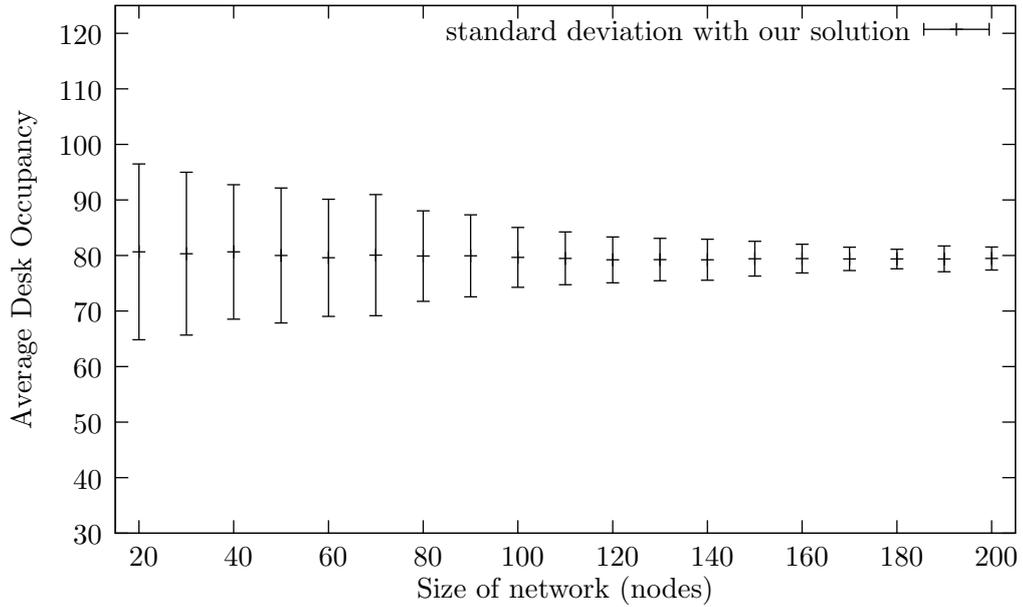


Figure 4.3: It shows the average desk occupancy over a two-hour period based of 15 minutes sampling. The occupancy results are presented in its accurate value (based on all 120 entries for each user) and the standard deviation resulting of 500 runs of our solution using 15-minute sampling. These results vary accordingly to the size of the network and they are based on the data being fully distributed (only the node themselves have access to the raw data).

on the target privacy protection for the data, with the trade-off solely defined by the data holder. In the second technique we propose the expansion of the concept of budgeting so that each queried element spreads the consumed amount of budget across either its neighbours or timed intervals. The spreading effect is directly linked with the chosen database sensitivity. The decision of using the latter is solely dependent on the data analyst, since the level of privacy remains exactly the same for the data holder.

Finally, this chapter has set the flexible and extensible groundwork that will form the basis for the framework for designing privacy-preserving data aggregation protocols presented in the next chapter.

Chapter 5

Framework for Designing Privacy-preserving Data Aggregation Protocols

The techniques presented in the previous chapter provide a strong and flexible set of tools to achieve privacy in a distributed time-series setting. So far, however, they have not been considered in the context of a wireless sensor network environment. This chapter presents how they could be weighted when designing a privacy-preserving data aggregation protocol for wireless sensor networks. It considers the key factors that influence the data collection and data storage, as well as data transmission in such networks.

The main target for the framework presented here is those designing the deployment of wireless sensor networks, single-purpose or multi-application network deployments, with the latter representing features made available as part of the middleware. Additionally, the framework is also expected to be used when designing applications to be deployed on existing networks, leveraging their multi-application support.

In wireless sensor networks, the end goal is to be able to answer questions regarding the sensed environment. When designing a privacy-preserving data aggregation protocol, the questions to be answered have to be clear and well defined. This does not mean, however, that these networks may not be used for anything else. It simply sets the benchmarks in terms of application needs, which will assist in defining the required privacy levels.

The influence to privacy levels and data utility come from architectural characteristics of the network, like topology and data storage, assumptions of the sensed environment, when it comes to placing the trust, as well as configurable characteristics, like defining data grouping and budget management metrics.

The next sections will expand on each of these influential factors. It will be followed by an overview of the framework in section 5.6.

It is worth mentioning upfront that none of the characteristics on the subjects covered in the coming sections can be represented in on or off fashion. Their applicability, most of the time, offer varying degrees of flexibility depending on the purpose of the network.

5.1 Placing the Trust

One of the first factors to be decided when designing a privacy preserving data aggregation algorithm is the threat model to be mitigated, which, in order words, means where to place the trust in the network.

As presented in chapter 2, wireless sensor networks are composed of several components which communicate in order to sense the environment. The interaction with the networks normally takes place via the sink, which acts as the network entry point for external stakeholders, like data analysts.

When placing the trust, a key factor is to decide whether the aimed protection is against external stakeholders or if it should be protected against other nodes too. Privacy constraints aside, the two options have quite contrasting effects in relation to data utility and noise levels.

In the former, data would travel from the nodes where they are collected all the way to the boundaries of the network via the sink. Once there, the data is aggregated and differential privacy is applied as shown in the previous chapter. This centralised model provides protection against the set difference attack because, from an external point of view, the isolation of nodes is not possible. The main concern with this option, however, is the inability of protecting the data from insiders or someone posing as one. The control of the data is far from where it is collected.

When the desire is to protect against other nodes, the control of the data is reclaimed from the sink and returned to the nodes. This distributed model provides a much stronger data privacy model because the differential privacy guarantee is applied by the node to its data. No data leaves the realms of the node unprotected. Of course, there is a cost on data utility when this model is adopted, restricting the granularity of queries that are possible. Chapter 6 has more insights into these limitations.

The control of the data, however, does not need to either be given to the sink or the nodes. As a way of benefiting from both options, clusters of nodes, possibly exploring some level of affinity among them, could be created so that we end up with a hybrid approach where each cluster operates in a centralised model internally and in a distributed model externally. From the point of view of a cluster, non-member nodes would be seen as ‘outsiders’ with regards to data privacy.

The decision on where to place the trust should not be taken in isolation. For example, the possibility of combining other techniques presented in section 5.5 could have a positive effect in favouring one model over another.

5.2 Network Topology

From a network topology point of view, there are three sets of aspects that could influence the design of a privacy-preserving data aggregation protocol using this framework: densely populated network versus small deployments (size), deep versus shallow topology and fixed versus mobile node locations.

The size of a deployment has a strong influence on the data utility of a network. The bigger the number of nodes, the more decentralised the control of data points can be. As we move towards smaller networks, we may end up having to centralise this control because, otherwise, the noise levels overshadow the sensed data. To some extent, the size of a deployment goes hand in hand with the decision on where to placing the trust in a network, as seen in section 5.1.

In the same way that the size of the deployment has an effect on what can be achieved in terms of data privacy, its topology also plays a relevant role. Deeper topology provides an easier to group organisation, therefore providing a higher utility and influencing the decision towards a hybrid trust model.

Likewise, for fixed node deployments, it is quite straightforward to accommodate grouping of devices and therefore play around with the trust model in favour of more appropriate solutions for the actual needs. Mobile deployments, on the other hand, can have an extra factor of complication for applying a hybrid trust model, favouring either a decentralised or a centralised model, unless some form of a dynamic clustering technique is employed.

5.3 Storage and Data Feeding Model

From a data storage point of view, a network could adopt two different approaches: historical data or live-data only. With regards to the feeding model, the options of operation are constant data feeder or on demand.

Networks with a certain level of storage allows for queries over historical data, which consequently provides data analysts with a wider range of possibilities, specially combined with the data grouping and budget management (see section 5.4 for more details). In a live-data only model, e.g. no local storage, queries can only be applied to whatever is sensed at the time of the query. Alternatively, when a hybrid or centralised approach is preferred, by the inclusion of storage nodes to the network, it is possible to convert a live-data only network into a historical data one, with the storage nodes acting as the cluster aggregator and therefore applying the exact amount of noise as expected.

When it comes to data feeding model, an application could be regularly publishing data, so effectively operating as a constant data feeder. The combination of a constant data feeder mode and budget management does not fit well, unless storage nodes are added to the network as described above, because it removes from data analysts the possibility of assigning different levels of budget depending on their actual requirement. In fact the combination of decentralised trust model, constant data feeding and no data storage does not go well together at all when designing a privacy-preserving data aggregation algorithm because it eliminates the possibilities of leveraging over configurable factors of the network.

In general, an on demand nature of the network provides a more granular level of possibilities to data analysts, especially as the number of analysts grow the number of applications expands because most likely the interest over the sensed data will be different, leading towards more specialised queries or use cases.

5.4 Application Needs vs Privacy Goals

Like for placing the trust, it is the application needs and privacy goals that essentially influences the configuring and decision making on the directions to take when it comes to data grouping and budget management. These two characteristics have been introduced in chapter 4.

5.4.1 Data Grouping

The level of data grouping defines how much of the data being generated by nodes or clusters of nodes will be treated as a single entry from the point of view of the definition of differential privacy. The options here vary from simple grouping of consecutive data points, like two hours or one day slots, or it could cover entries spanning across non-consecutive sequences, like the first of every hour in a day. The decision on the grouping depends on the level of correlation of the events being sensed. A combination of several groups is also possible, like grouping the data into hour slots as well as these hour slots into a whole weeks.

The way data grouping is set should be heavily influenced by nature of the data and the environment where it is being collected. That inline with the privacy goal and application needs should drive how it is configured.

5.4.2 Budget Management

In addition to choosing a data grouping policy, it is also necessary to define the amount of budget available for queries. As previously mentioned, each query consumes a certain amount of this privacy budget. This behaviour creates one of the most interesting aspects of the framework because it gives to external elements of the networks, e.g. data analysts or users of the network, the capability of deciding how much budget to be consumed with each query. This means that it is up to the users of the network to choose whether they concentrate the amount of budget allocated to them into one particular data point within the grouped data, spread it equally across all data points, or opt for a subset of the data.

Equally like the data grouping aspect, it is possible to define an overall amount of budget for a group of slots as a way of protecting more correlated data elements. For example an application could define a certain budget for a data grouping slot of two hours and, in addition, define a budget limit for the day. This would mean that not all 2-hour slots could have their budget fully consumed. In summary, it would be possible to create layers of budget, with the outer layer always smaller than the sum of the allocated budget of all elements of the adjacent inner layer.

Most likely on multi-application deployments, nodes or clusters of nodes could have the budget of their data groups saturated by several applications. Therefore, a policy for budget consumption across application could be put in place to prioritise or apply more complex allocations. On its simplest form, a first come first serve model could be applied.

The decision on the amount of budget to allocate to each node is application and data specific, given that the budget directly represents the amount of privacy being traded in exchange for data utility.

5.5 Data Perturbation in Conjunction with Other Techniques

It is possible to combine the techniques presented in chapter 2 with the techniques presented so far in this chapter to achieve higher levels of privacy without affecting noise levels. As demonstrated in chapter 3, on their own, the former could not withstand the set difference attack. However, as an addition to the latter, it can contribute towards higher privacy guarantees without compromising utility since it does not incur noise.

5.5.1 Clustering

This method of data aggregation combines the data points of neighbouring nodes and is directly related to the concept of clustering proposed in section 5.1 when discussing where to place the trust in a network.

The protocol here would almost work in the same way the original protocol works as described in chapter 2, with the different that after the second round, the aggregator applies the expected level of differential privacy noise to the result before transmitting it towards the sink of the network.

The motivation for adopting this model comes from the idea of trusting the nodes nearby could be more acceptable, especially considering the less noisy results that it achieves. Of course, in contrast, it is relevant to highlight that by adopting it, the system will still be susceptible to a variation of the set difference attack. It is a variation because the attacker has to be another node (inside the cluster) and not anyone that poses as a data analyst sending queries to a wireless sensor network application as in the original attack. What needs to be remembered, however, is that this protocol is only applicable when not using the fully distributed trust model, which, inherently, means that the data has to be transmitted to the aggregator anyway, so, by adding this protocol, it would be better off, despite being susceptible to a much milder variation of the set difference attack.

5.5.2 Slicing

This technique basically unevenly splits the data to be transmitted into two or more parts and sends them towards the sink via different paths. Its use does not seem too helpful when the data in question is already differentially private. It can make a difference, however, when the trust model in use is a hybrid or centralised one. Using this technique, we would protect the sensed data during transmission towards the aggregator, minimising the possibility of data interception from insiders.

5.5.3 Other Techniques

Other techniques could be applied to varying parts of the network to mitigate attacks or at least reduce their potential against data leakage. One example of such techniques would be the use of homomorphic encryption, presented in section 2.6.4.

5.6 Framework Overview

This section presents a summary of the characteristics presented so far to assist in designing a privacy-preserving data aggregation protocol for wireless sensor networks. We start by summarising all characteristics that will be classified from the point of view of data utility, point of control, targeted protection and finally configurability.

The characteristics presented as part of the framework are:

- Placing the trust: distributed, centralised or hybrid approaches
- Network size: number of nodes sensing the environment
- Network topology: deep versus shallow topology
- Network mobility: fixed versus mobile nodes
- Data Storage: support for historical data versus live data only
- Feeding model: constant versus on demand
- Data grouping: defines the blocks of data to be seen as one from the differential privacy point of view
- Budget management: defines how the consumption of privacy budget takes place

- Conjunction of data aggregation techniques: use of existing data aggregation techniques like clustering

The first criteria for classifying these characteristics is how they behave from the point of view of data utility. Therefore, the classification is under one of three options: increase, decrease or not applicable. The classification is presented in table 5.1.

Data utility			
characteristic	increases	decreases	n/a
Placing the trust	centralised/hybrid	distributed	
Network size	big	small	
Network topology	deep	shallow	
Network mobility	fixed	mobile	
Data storage	historical data	live data only	
Feeding model	on demand	constant	
Data grouping	smaller groups	bigger groups	
Budget mgmt	✓		
Other techniques			✓

Table 5.1: This table presents the classification of the characteristics of the framework with regards to data utility.

Another classification is from the point of view of control, defining who has the power to set it. The possible options under this classification are: external (data analysts/users), network, node level and not applicable. The classified characteristics is presented in table 5.2.

Point of control				
characteristic	external	network	node-level	n/a
Placing the trust		centralised/hybrid	distributed	
Network size				✓
Network topology				✓
Network mobility				✓
Data storage		✓	✓	
Feeding model		✓	✓	
Data grouping		✓	✓	
Budget mgmt	✓			
Other techniques				✓

Table 5.2: This table presents the classification of the characteristics of the framework with regards to point of control.

From the point of view of targeted protection, the classification accepts the following options: insiders, outsiders and not applicable. Table 5.3 contains the classified characteristics.

Targeted protection			
characteristic	insiders	outsiders	n/a
Placing the trust	distributed/hybrid	centralised/distributed/hybrid	
Network size			✓
Network topology			✓
Network mobility			✓
Data storage			✓
Feeding model			✓
Data grouping	✓	✓	
Budget mgmt			✓
Other techniques	✓		

Table 5.3: This table presents the classification of the characteristics of the framework with regards to targeted protection.

Finally, we present the classification of the above characteristics with respect to their configurability. Possible values are configurable and network aspect. The former, as the name states, can be chosen while the latter is a given that simply acts as an influential factor for other decisions. This classification can be found in table 5.4.

Configurability		
characteristic	configurable	network aspect
Placing the trust	✓	
Network size		✓
Network topology		✓
Network mobility		✓
Data storage		✓
Feeding model		✓
Data grouping	✓	
Budget mgmt	✓	
Other techniques		✓

Table 5.4: This table presents the classification of the characteristics of the framework with regards to its configurability.

These classifications have been set from the point of view of a protocol being designed for an existing network, therefore assuming that the network aspects are given characteristics. If the network still does not exist, then table 5.1 should be taken into account when deciding on the network aspects.

5.7 Conclusion

This chapter presents a framework for designing privacy-preserving data aggregation protocols for wireless sensor networks. It combines the proposed techniques for performing data perturbation to distributed time-series data presented in the previous chapter with relevant aspects of wireless sensor networks like topology and network sizes.

Furthermore, the framework does not force the design of privacy-preserving protocols in any particular direction. The driving factor should always be the data utility should be inline with the privacy expectations. For example, if deciding for maximum privacy in a small deployment, wide data grouping, among others, the network will be of very little to no utility.

Finally, the framework remains open for expansion, so that new techniques can be incorporated, further enhancing its capability of extracting maximum utility from a wider range of scenarios.

Chapter 6

Evaluation

The combination of the characteristics, described in the previous chapter as part of the framework, can have a profound effect on applications running in wireless sensor network. Thus, this chapter aims to demonstrate such an effect by presenting two extreme cases of applying the framework followed by a hybrid approach that uses a more balanced combinations of factors.

This chapter is organised as follows. Section 6.1 sets the scene that will be used in the evaluations. In sections 6.2 and 6.3, we evaluate the effects of running a centralised and a decentralised trust model respectively. Next, a more well-balanced approach is evaluated, adopting a hybrid trust model, in section 6.4. The chapter concludes with section 6.5.

6.1 Approach

The evaluations are going to mainly focus on the placing of the trust factor since, as presented in section 5.6, it is a configurable characteristic that causes a big impact in data utility and it can be used for shifting the point of control and leveraging the protection of the data against insiders or outsiders.

It has been mentioned several times that it comes down to the application needs as well as privacy requirements the decisions of how to design a suitable privacy-preserving protocol. That statement continues to be accurate. The goal here, however, is slightly different. These evaluations are exploring the capability of the framework of achieving data utility by designing two extremes cases and one more likely to be adopted. Therefore, these evaluations are orthogonal to the application in question. The other factors, like network topology and data storage, being network

aspects as shown in table 5.4, are going to influence the decision for the trust model to be used.

For the evaluations, a data set has been created to represent occupancy events in our office occupancy case study. The data points take the form of Boolean entries that have been simulated to be collected every minute for a period of one hour. This data set could represent any somehow detectable event, like movements, a threshold being reached, or even the explicit activation of a switch.

The data will be grouped into one hour chunks, therefore setting the granularity of the database to 60, which is the total number of collected data points in one hour. This setting means that the entire hour should be considered as one from the point of view of fulfilling the differential privacy definition presented in chapter 4.

6.2 Centralised Trust Model

This evaluation is concerned with analysing the data utility of the centralised trust model. In order to achieve that, a simulation has been conducted to replicate the behaviour of the network as the number of nodes in the network increases. This particular simulation covers network sizes between 20 and 200 nodes.

The resulting outcome of the simulation is going to be presented in the form of three graphs, which plot the relationship between the average number of events taken place within a particular timestamp and the number of nodes in the network. For comparison, each graph assumes a different level of budget expenditure: 5, 1 and 0.2. The graphs also include the standard deviation after 500 runs. Figures 6.1, 6.2 and 6.3 present the results of this simulation.

The results of the simulation matches the expectation that a centralised trust model achieves high levels of data utility. The convergence of the answer when consuming budget at levels 5 and 1 is high even for small deployments. For level 0.2, however, the range is prohibitive high for small deployments. As the network grows, the accuracy improves.

This evaluation makes a reasonable assumption that the data sent by the nodes is stored by the sink. This assumption can be justified by the fact that it is reasonably straight forward to achieve that given its central location. In turn, with this assumption, the feeding model allows for on-demand type of queries.

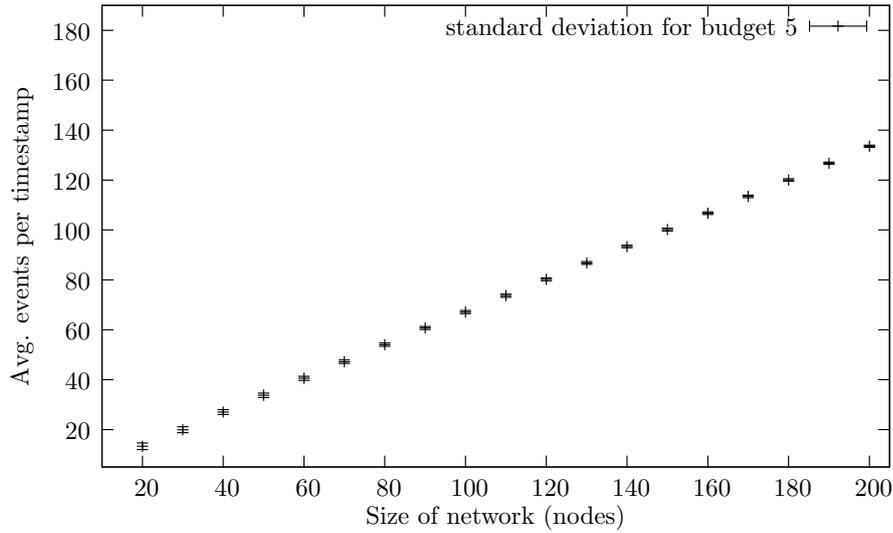


Figure 6.1: It presents the behaviour of a network that adopts a centralised trust model with budget expenditure 5.

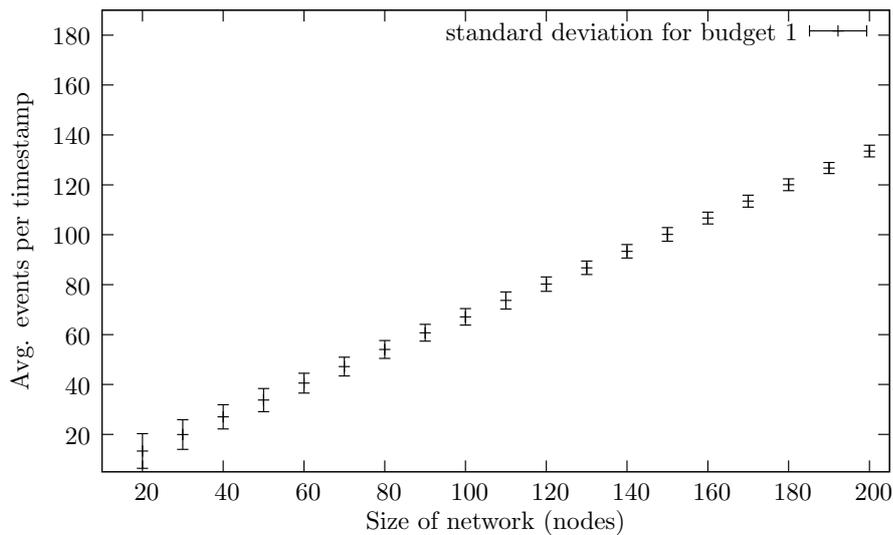


Figure 6.2: It presents the behaviour of a network that adopts a centralised trust model with budget expenditure 1.

6.3 Decentralised Trust Model

This evaluation focuses in analysing the data utility levels of the decentralised trust model and follows the same format as the previous one. The only differences are a consequence of the noisier nature of the decentralised trust model. For example, the network sizes used in the simulations vary from 50 to 1000 nodes, considerably bigger

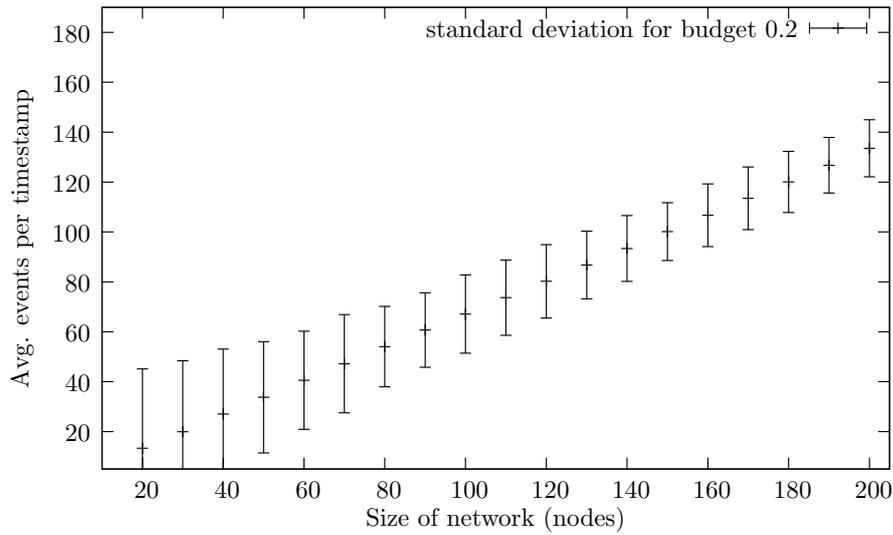


Figure 6.3: It presents the behaviour of a network that adopts a centralised trust model with budget expenditure 0.2.

networks compared to the 20 to 200 nodes used for simulating the centralised trust model. Another difference is with regards to the amount of budget. We have replaced the 0.2 level with the 0.5 one because of the meaningless expressiveness of the former. The results of this simulation, after 500 runs each, are presented in Figures 6.4, 6.5 and 6.6.

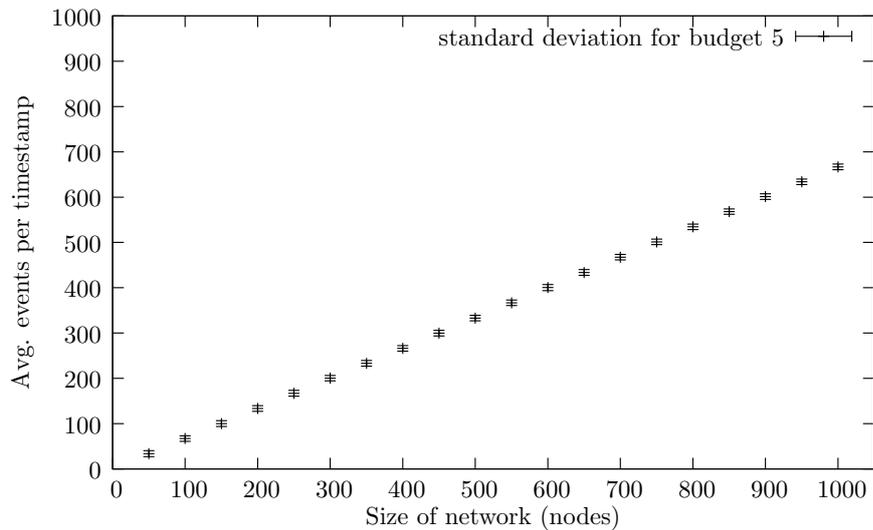


Figure 6.4: It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 5.

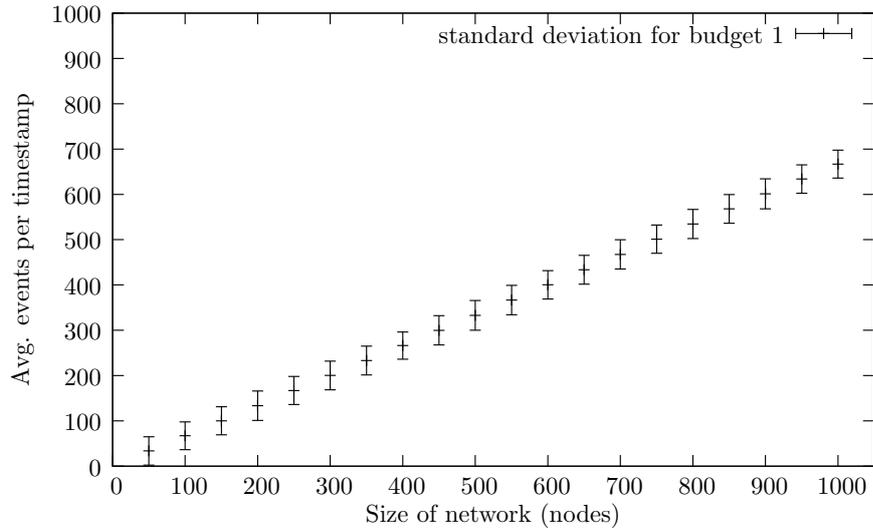


Figure 6.5: It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 1.

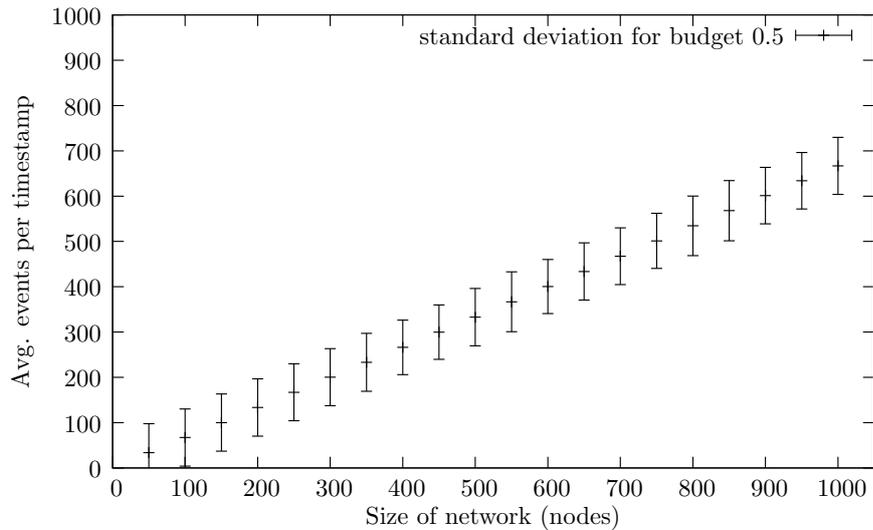


Figure 6.6: It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 0.5.

As expected, the adoption of a decentralised trust model results in considerably noisier outcomes, especially compared to the centralised trust model. While for the results with the consumed budget at level 5, the accuracy is quite high from very early on, the results for budget level 0.5 present a very high standard deviation throughout.

It is not feasible to run a decentralised trust model without local data storage because the level of perturbation would simply overshadow the actual data, similar

to the behaviour caused when using budget consumption level at 0.1. Therefore, this experiment worked on the assumption that nodes store the sensed values so that it can later be queried.

Given the noisier characteristics of the decentralised trust model, there is a possibility of querying a larger dataset than that of the data group. So, we have incremented the dataset with additional data and run another experiment. To be more precise, we have loaded four hours worth of data. The rationale behind this attempt is that we are going to include more actual data into the response without adding any extra amount of noise. That is because we have already covered the noise levels to fulfil a data group and the extra data is coming from data entries that are part of other data groups. The result of this experiment can be seen in Figures 6.7, 6.8 and 6.9.

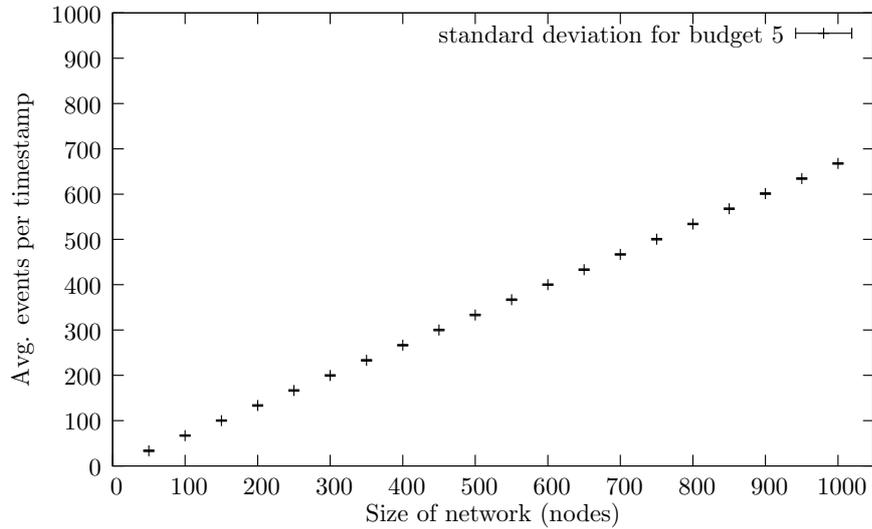


Figure 6.7: It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 5. The questions cover 4 times more data points than the size of the data grouping in use. For this particular experiment it used 240 data points from each node since the data grouping size is 60 data points.

The results found in Figures 6.7, 6.8 and 6.9 are a great improvement when compared to the previous experiment. Even for the 0.5 budget expenditure level there is a better convergence towards the true answer. The impact, however, is that our average now covers a much wider date range, with four hours worth of data. Alternatively, another possibility for increasing data utility is to make use of the budget management feature and query only a subset of the available data points in a data group. That way, we could trade much better data utility levels in exchange for a less

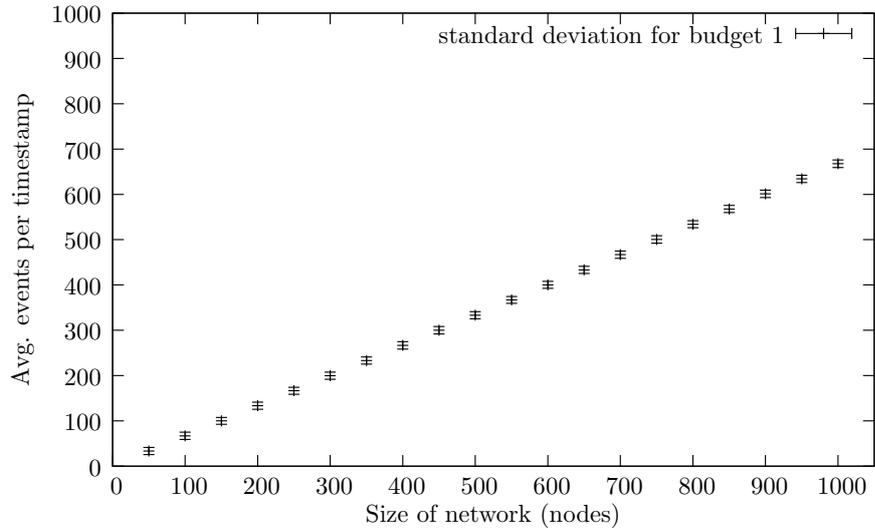


Figure 6.8: It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 1. The questions cover 4 times more data points than the size of the data grouping in use. For this particular experiment it used 240 data points from each node since the data grouping size is 60 data points.

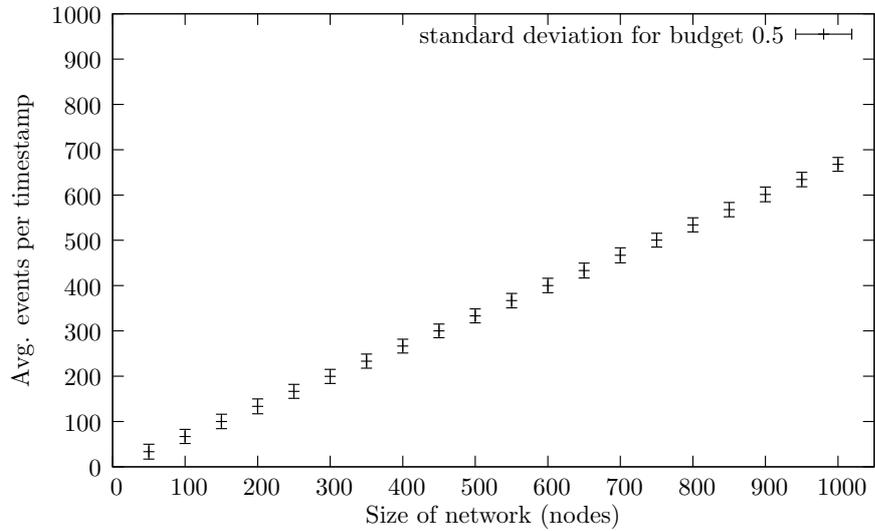


Figure 6.9: It presents the behaviour of a network that adopts a decentralised trust model with budget expenditure 0.5. The questions cover 4 times more data points than the size of the data grouping in use. For this particular experiment it used 240 data points from each node since the data grouping size is 60 data points.

granular dataset. Additionally, we would include the same subset of data points for other data groups, resulting in better data utility levels as demonstrated in Figures 6.7, 6.8 and 6.9.

6.4 Hybrid Trust Model

This evaluation focuses in analysing the data utility levels of a hybrid trust model and follows the same format as the previous ones. The obvious difference is the existence of clusters of nodes, which, for this experiment, has been set to ten clusters. In addition, the number of nodes being experimented on varies between 25 and 295. Finally, the budget consumption levels are back to the values used in the centralised trust model experiment. The experiments have been performed 500 times. The results of this simulation are presented in Figures 6.10, 6.11 and 6.12.

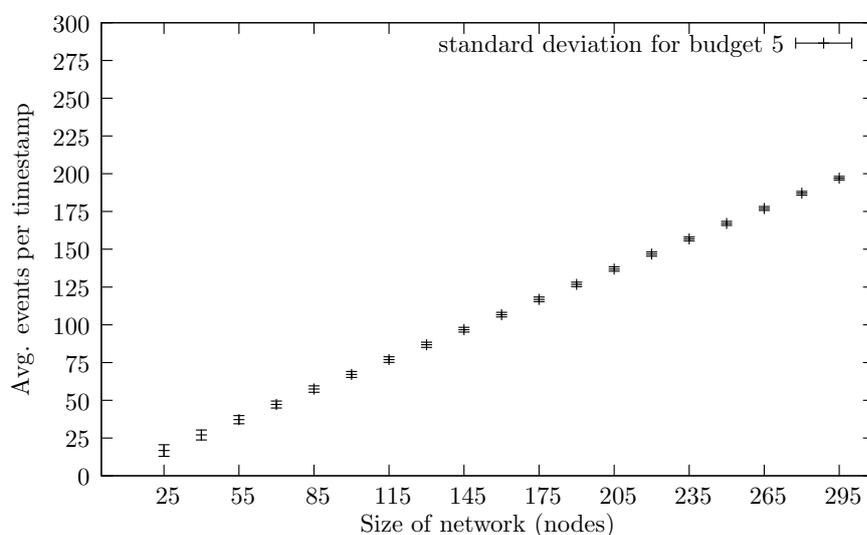


Figure 6.10: It presents the behaviour of a network that adopted a hybrid trust model with 10 clusters of nodes and budget expenditure 5.

The results of Figures 6.10, 6.11 and 6.12 show an expected outcome from the data utility point of view. It performs considerably better than the decentralised trust model but worse than the centralised one. The surprising result is with the 0.2 budget consumption level (Figure 6.12), which shows very noisy results for small deployments.

In addition, there is a second experiment that attempts to evaluate the impact that the number of clusters can have in a hybrid deployment. It uses budget consumption level 1 with cluster sizes 10, 50 and 100 clusters presented, respectively, in Figures 6.13, 6.14 and 6.15.

Figure 6.13 shows that as the number of cluster gets smaller, 10 in this case, so does the similarities to a centralised trust model in term of data utility. The opposite

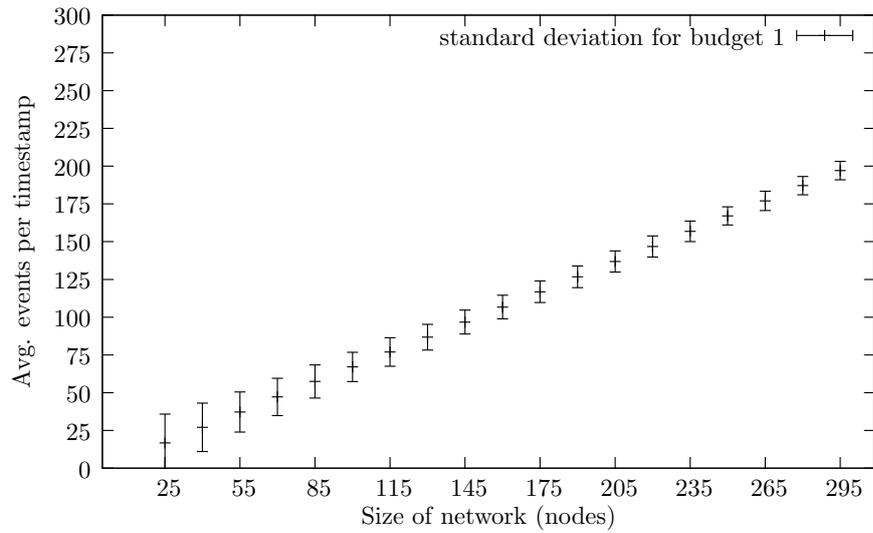


Figure 6.11: It presents the behaviour of a network that adopted a hybrid trust model with 10 clusters of nodes and budget expenditure 1.

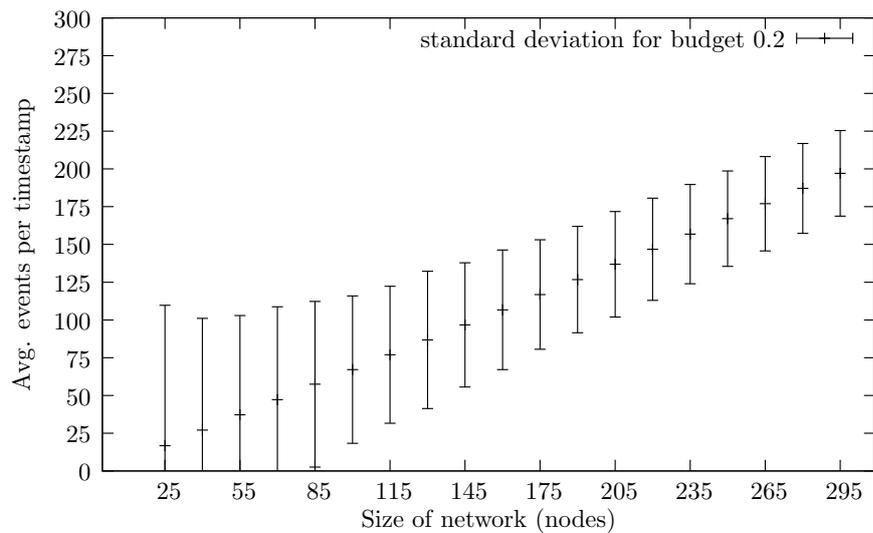


Figure 6.12: It presents the behaviour of a network that adopted a hybrid trust model with 10 clusters of nodes and budget expenditure 0.2.

is also true, as the number of cluster grows (100 clusters in Figure 6.15), bigger is the correspondence to a distributed trust model.

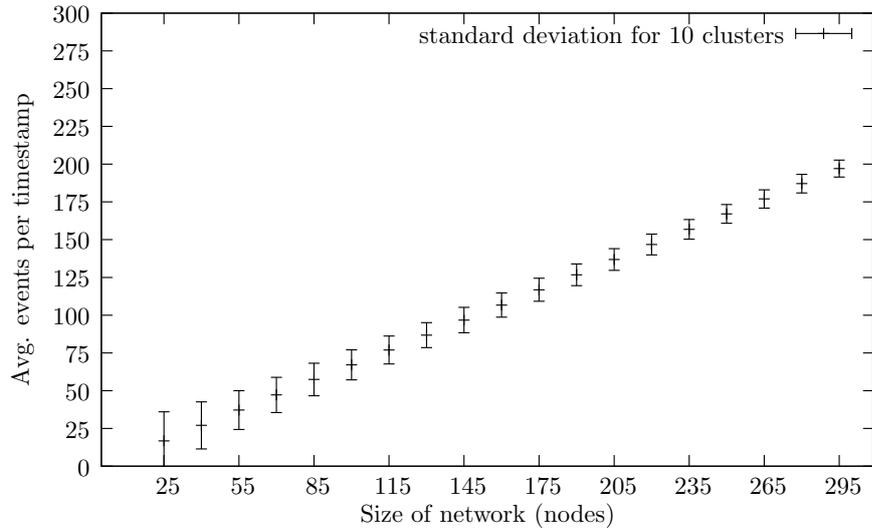


Figure 6.13: It compares the behaviour of the network that adopted a hybrid trust model, from the point of view of data utility, for a total of 10 clusters.

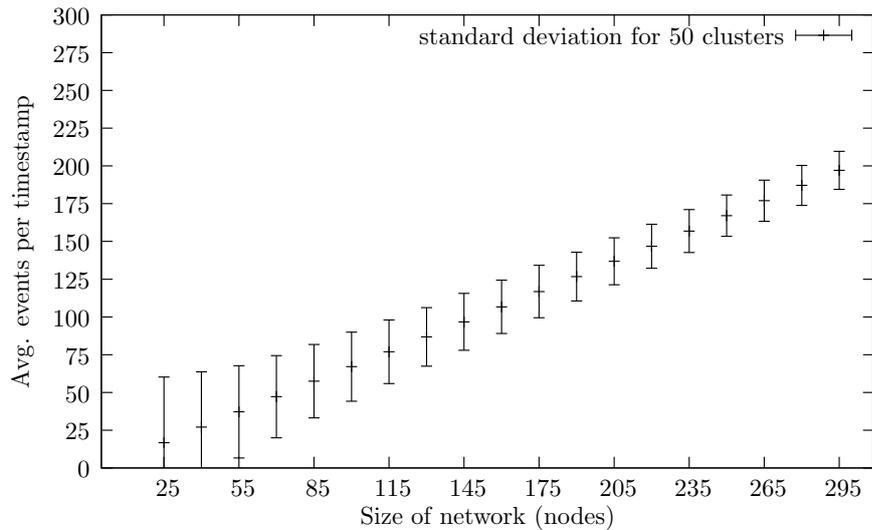


Figure 6.14: It compares the behaviour of the network that adopted a hybrid trust model, from the point of view of data utility, for a total of 50 clusters.

6.5 Conclusion

In this chapter, we compare and contrast the use of several characteristics of the framework proposed in chapter 5 to address data privacy in wireless sensor networks. The first two evaluations employ opposite extremes to deal with data privacy. The first relies on a centralised trust model, therefore focusing on protecting against ex-

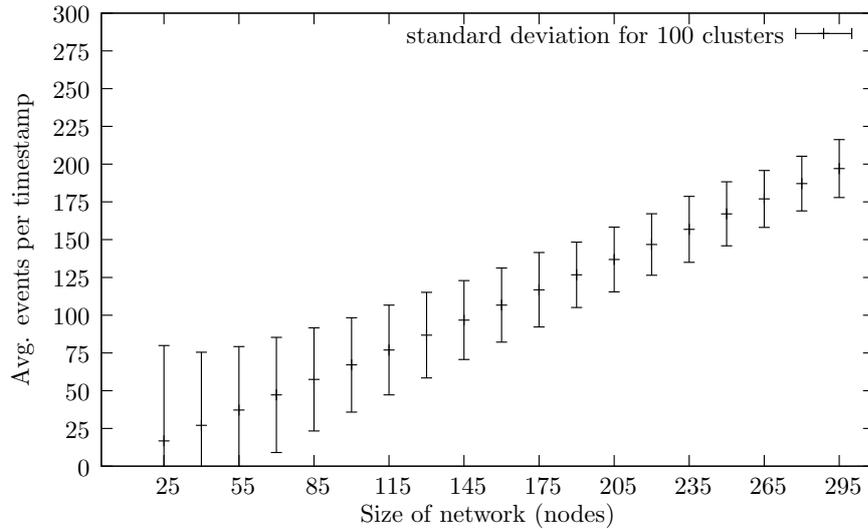


Figure 6.15: It compares the behaviour of the network that adopted a hybrid trust model, from the point of view of data utility, for a total of 100 clusters.

ternal elements of the network. The second set of evaluations, on the other hand, aims for a maximum level of privacy from the point of view of control of the data, adopting a decentralised approach which gives the control over the data to the nodes. The last set of evaluations aims to benefit from the higher levels of data utility of the former while relaxing the level of privacy of the latter, adopting a hybrid trust model.

Among the results presented in the simulations, there are some obvious ones like the higher data utility found in centralised protocols when compared to their decentralised counterpart. It is important to mention, however, that the extent of the reduced level of data utility of the latter is quite expressive, especially for smaller networks. The budget consumption levels also had to be tweaked due to the inexpressiveness of the results achieved.

It has been constantly highlighted and should also be noticeable that there is no necessarily right or wrong answer when it comes to designing privacy-preserving data aggregation protocols. What in fact exist are sets of requirements from the different stakeholders involved or affected by a deployment or a particular application. Inadvertently, a higher level of privacy is going to lead to smaller levels of data utility, especially as the point of control of the network is shifted towards the nodes [49]. The challenge is to find the correct balance between data utility and data privacy.

Chapter 7

Conclusion and Future Work

This dissertation has investigated the extent to which data perturbation is a suitable mechanism for addressing privacy concerns in wireless sensor networks. During this research, several contributions have been made to provide a better understanding as well as towards techniques to leverage acceptable levels of data utility and data privacy. These contributions are discussed in section 7.1.

Additionally, section 7.2 outlines several future opportunities and open problems to explore. Finally, section 7.3 concludes by summarising the contributions of the dissertation and answering the thesis question.

7.1 Contributions

This section presents the contributions of this dissertation. They have been grouped into three main areas of interest.

7.1.1 Unsuitability of Existing Privacy-preserving Protocols in Multi-application Wireless Sensor Networks

This dissertation has analysed how existing privacy-preserving protocols perform on multi-application networks and it has shown that their performance do not match their proposed goals. The protocols mainly obfuscate the data in transit but fail to address more obvious attack models.

In order to demonstrate the unsuitability of these protocols, the set difference attack has been proposed. The attack exploits the intersections between the sets of sensors comprising applications to discover scenarios in which individual nodes, or

small clusters of nodes, are isolated. Interestingly, nodes that are under attack have no way of preventing or even detecting the attack.

In investigating potential ways of mitigating the attack, it was informally identified that the lack of any form of uncertainty plays a key role in making the attack succeed, projecting towards the need for a formal data perturbation as the means to achieve data privacy in wireless sensor networks.

7.1.2 Formally Achieving Privacy in Wireless Sensor Networks

We have shown that, although differential privacy has been a breakthrough to the statistical privacy database, it falls short for time-series data, which by itself is not an entirely new finding, but it also reveals that there are applications that are using the current definition of differential privacy indiscriminately, ignoring its limitations. Such limitation has been demonstrated by the effectiveness of the statistical set difference attack, which explores the granularity misalignment between datasets and the aimed levels of privacy.

In order to mitigate the attack, two approaches have been proposed. The first introduces the concept of database sensitivity, which extends the privacy protection to wider ranges of data points. The sensitivity level is configurable and depends directly on the targeted privacy protection for the data, with the trade-off solely defined by the data holder. The second technique proposes the expansion of the concept of budgeting so that each queried element spreads the consumed amount of budget across either its neighbouring or timed-interval data points. The spreading effect is directly linked to the chosen database sensitivity. The decision of using the latter is solely dependent on the data analyst, since the level of privacy remains exactly the same for the data holder.

7.1.3 Framework for Designing Privacy-preserving Data Aggregation Protocols

The final set of contributions comes from the framework devised to assist in the design of privacy-preserving data aggregation protocols for wireless sensor networks. The framework presents a comprehensive set of characteristics to be taken into account and offers a wide variety of configurable options. At its core is the flexibility and non-imposition of any aspect onto the network.

One relevant aspect of the framework is that it allows the utilisation of existing privacy-preserving protocols in ways that emphasise their real strengths, providing an interesting alternative for networks opting for a hybrid or centralised trust model.

Succinctly, the framework acts as a guideline to assist in extracting the most data utility from the network while maintaining an acceptable level of data privacy.

In order to evaluate the framework, simulations have been conducted to demonstrate the suitability of the framework to address contrasting goals from the point of view of data privacy and data utility. The first evaluation, adopting a centralised trusting model, achieves high accuracy levels, even for a small deployment and, from the point of view of external entities of the network, it provides strong privacy guarantees. The second set of simulations, adopting a decentralised trusting model, contrasts with the findings of the first evaluation because it requires a much larger network deployment with queries covering a wide range of data points in order to achieve some levels of data utility. The final set of simulations leverages several characteristics of the framework to be more in line with real deployments. It adopts a hybrid trust model and performs considerably better than the decentralised one. Variations in the number of clusters is also evaluated.

7.2 Future Work

In addition to the considerations presented above, this dissertation has set the foundation that enables several avenues for future work. These have been grouped into four categories.

7.2.1 Enhancement of the Framework

As mention in section 7.1.3, the proposed framework has set a flexible environment around the designing of privacy-preserving data aggregation protocols. Despite its novelty, it has only set the foundation, with a lot more to be explored.

Among the areas to be expanded are the incorporation of new techniques to provide for stronger trust models, enhancing the understanding of the effect of different architecture in achieved data utility and privacy levels, further exploring the composition capabilities of the framework to further extend its applicability, etc.

Another interesting aspect to be incorporated into the framework is the support for algorithms that retain their privacy properties even if their internal state becomes

visible to an adversary, known as Pan-Private algorithms [30], which is highly relevant in the context of wireless sensor networks and their distributed node deployment.

7.2.2 Improving Data Utility

The main means of achieving privacy employed in this research has been through the addition of perturbation to answers, which means that the noisier the answer, the less is the extent of its utility and vice versa.

Therefore, it is of great interest to explore new means of reducing the level of noise and improve the utility of the network without necessarily compromising on privacy. One potential avenue could be the exploration of the imprecision of sensors in use [82]. This would result in less explicit noise having to be added to the answers while still fulfilling the proposed relaxed definition of differential privacy.

Another direction for improving the noise levels would be the investigation of more optimal ways of generating distributed noise to fulfil differential privacy in the context of wireless sensor networks. Improvements have already been made for different environment settings [29, 13].

Further improvements can also be directed towards the organisation of the networks. In a centralised model, the employment of homomorphic encryption [37] would improve the privacy levels while achieving the high utility levels that such a model entails. In relation to the distributed model, techniques like those employed in distributed analytics [40, 12] could boost data utility while maintaining the high levels of privacy that is characteristic of this model.

7.2.3 Privacy by Design

As mentioned earlier, the framework for designing privacy-preserving protocols in wireless sensor networks is one of the main contributions of this research. However, another (unexpected and interesting) outcome of the framework which could be explored in further research is its utility as a tool for achieving privacy by design.

By changing the viewpoint of the framework, instead of acting on existing deployments, it enables the design and implementation of wireless sensor networks taking into account the privacy by design principles set by Gürses et al. [39]. This is only possible due to the open and flexible characteristics of the framework, which does not impose any particular factor onto its users. Researching the extent of this relation and further exploring the alignment with the principles could result in a relevant

outcome, especially towards making privacy at the heart of privacy-sensitive wireless sensor networks and not an afterthought.

7.2.4 Mobile Network

This research has mainly focused on the context of wireless sensor networks. An interesting direction of research could be the investigation on how these findings would behave in the context of mobile networks, more specifically mobile phones. Mobiles phones are a ubiquitous part of our society, which makes the research even more interesting and with overreaching results.

Despite presenting some similar characteristics, like time-series and distributed data, such investigation presents additional challenges, like a more heterogeneous environment, a business model that incentivises a hungriness for more and more data, etc. [51, 19, 48]. Nonetheless, the contributions of this dissertation could be widely applicable to such environment, which can further benefit from ubiquitous internet connectivity and consequently easier grouping of data from individuals via the employing of social networking features.

7.3 Summary

This dissertation has set out to investigate how the shift in the organisation of a wireless sensor network affects data privacy as a whole, ranging from the point of view of the user being subjected to the sensing environment to that of the network owner.

To answer the thesis question, the first part of this dissertation has demonstrated how the shift in paradigm guarantees the effectiveness of the set different attack. The second part has built upon the first by presenting how it is possible to formalise data perturbation in the context of wireless sensor networks. The third and final part has combined the results of the first two parts and has presented a comprehensive and flexible framework for designing privacy-preserving protocols that gives network owners as well as those being subject to the network the capability to protect their privacy. The framework has been evaluated, through experiments, to demonstrate the achievable levels of data utility in a variety of network setting.

It also includes the experiments to evaluate the data-utility levels that is achievable using framework.

For those reasons, this dissertation has succeeded in answering the thesis question and, in doing so, it has made several contributions and enabled the possibility for several avenues of future work.

Bibliography

- [1] Uday Acharya and Mohamed Younis. An Approach for Increasing Base-Station Anonymity in Sensor Networks. *2009 IEEE International Conference on Communications*, pages 1–5, June 2009.
- [2] Nabil R. Adam and John C. Worthmann. Security-control methods for statistical databases: a comparative study. *ACM Computing Surveys (CSUR)*, 21(4):515–556, 1989.
- [3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, March 2002.
- [4] Julia Albath and Sanjay Madria. Secure Hierarchical Data Aggregation in Wireless Sensor Networks. *2009 IEEE Wireless Communications and Networking Conference*, pages 1–6, April 2009.
- [5] Hani Alzaid, Ernest Foo, and Juan Gonzales Nieto. Secure data aggregation in wireless sensor network: a survey. In *Proceedings of the sixth Australasian conference on Information security*, volume 81, pages 93–105, Wollongong, NSW, Australia, 2008. Australian Computer Society, Inc.
- [6] Archana Bharathidasan and Vijay Anand Sai Ponduru. Sensor networks: an overview. *IEEE Potentials*, 22(2):20–23, April 2003.
- [7] Rabindra Bista and Jae-Woo Chang. Privacy-Preserving Data Aggregation Protocols for Wireless Sensor Networks: A Survey. *Sensors*, 10(5):4577–4601, May 2010.
- [8] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 128–138. ACM, 2005.

- [9] Athanassios Boulis, Saurabh Ganeriwal, and Mani B. Srivastava. Aggregation in sensor networks: an energy-accuracy trade-off. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 2003.*, pages 128–138. Ieee, 2003.
- [10] Olutayo Boyinbode, Hanh Le, Audrey Mbogho, Makoto Takizawa, and Ravi Poliah. A Survey on Clustering Algorithms for Wireless Sensor Networks. *2010 13th International Conference on Network-Based Information Systems*, pages 358–364, September 2010.
- [11] Haowen Chan, Adrian Perrig, and Dawn Song. Secure hierarchical in-network aggregation in sensor networks. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 278–287. ACM, 2006.
- [12] Ruichuan Chen, Istemi Ekin Akkus, and Paul Francis. Splitx: High-performance private analytics. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 315–326, New York, NY, USA, 2013. ACM.
- [13] Ruichuan Chen, Alexey Reznichenko, Paul Francis, and Johannes Gehrke. Towards statistical queries over distributed private user data. In *Presented as part of the 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 169–182, San Jose, CA, 2012. USENIX.
- [14] William Conner, Tarek Abdelzaher, and K. Nahrstedt. Using data aggregation to prevent traffic analysis in wireless sensor networks. *Distributed Computing in Sensor Systems*, pages 202–217, 2006.
- [15] John T. Correll. Igloo White. *Air Force Magazine*, pages 56–61, November 2004.
- [16] Carlo Curino, Matteo Giani, Marco Giorgetta, Alessandro Giusti, A.L. Murphy, and G.P. Picco. Tinylime: Bridging mobile and sensor networks through middleware. In *Third IEEE International Conference on Pervasive Computing and Communications, PerCom*, number PerCom, pages 61–72. IEEE Computer Society, 2005.
- [17] Tore Dalenius. Towards a methodology for statistical disclosure control. In *Statistik Tidskrift 15*, 1977.
- [18] George Danezis and Markulf Kohlweiss. Differentially private billing with rebates. *Information Hiding*, 2011.

- [19] Yves-Alexandre de Montjoye, Cesar A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the Crowd: The privacy bounds of human mobility. *Sci. Rep.*, 3, March 2013.
- [20] Dorothy E. Denning, Peter J. Denning, and Mayer D. Schwartz. The Tracker: A Threat to Statistical Database Security. *ACM Transactions on Database Systems*, 4(1):76–96, March 1979.
- [21] Dorothy E. Denning and Jan Schlörer. A fast procedure for finding a tracker in a statistical database. *ACM Transactions on Database Systems*, 5(1):88–102, March 1980.
- [22] T. Dimitriou. Secure Hierarchical Communications in Distributed Sensor Networks. In *Mobile and Wireless Communications Summit, 2007. 16th IST*, pages 1–6. IEEE, 2007.
- [23] Irit Dinur and K. Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210. ACM, 2003.
- [24] Yitao Duan, Netease Youdao, John Canny, and Justin Zhan. P4P: Practical Large-Scale Privacy-Preserving Distributed Computation Robust against Malicious Users. In *In The 19th USENIX Security Symposium*, Washington D.C., 2010.
- [25] Cynthia Dwork. Differential privacy. *Automata, languages and programming*, page 371, 2006.
- [26] Cynthia Dwork. Differential privacy. In *Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin / Heidelberg, 2006.
- [27] Cynthia Dwork. Differential privacy: A survey of results. *Theory and Applications of Models of Computation*, pages 1–19, 2008.
- [28] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.
- [29] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. *Advances in Cryptology-EUROCRYPT 2006*, pages 486–503, 2006.

- [30] Cynthia Dwork, Moni Naor, Toniann Pitassi, G.N. Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. *Manuscript submitted for publication*, 2009.
- [31] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 715–724. ACM, 2010.
- [32] Cynthia Dwork and Kobbi Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Advances in Cryptology CRYPTO 2004*, pages 134–138. Springer, 2004.
- [33] P. Erdős and A. Rényi. On a classical problem of probability theory. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 6:215–220, 1961.
- [34] European Commission. 95/46/EC-Data Protection Directive. *Official Journal of the European Communities*, 281:0031 – 0050, 1995.
- [35] Pierre-Alain Fouque, Guillaume Poupard, and Jacques Stern. Sharing decryption in the context of voting or lotteries. In *International Conference on Financial Cryptography*. LNCC, 2000.
- [36] Raghu K. Ganti, Nam Pham, Yu-En Tsai, and Tarek F. Abdelzaher. PoolView: stream privacy for grassroots participatory sensing. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 281–294. ACM, 2008.
- [37] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.
- [38] Ceki Gülcü and Gene Tsudik. Mixing email with BABEL. In *Symposium on Network and Distributed System Security*, pages 2–16, 1996.
- [39] Seda Gürses, Carmela Troncoso, and Claudia Diaz. Engineering Privacy by Design. In *Conference on Computers, Privacy & Data Protection*, August 2011.
- [40] Hamed Haddadi, Richard Mortier, Steven Hand, Ian Brown, Eiko Yoneki, Derek McAuley, and Jon Crowcroft. Privacy analytics. In *ACM SIGCOMM Computer Communication Review*, 2012.

- [41] S. Hadim and N. Mohamed. Middleware: Middleware Challenges and Approaches for Wireless Sensor Networks. *IEEE Distributed Systems Online*, 7(3):1–1, March 2006.
- [42] Wenbo He, Xue Liu, Hoang Nguyen, Klara Nahrstedt, and Tarek Abdelzaher. PDA: Privacy-Preserving Data Aggregation in Wireless Sensor Networks. In *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*, pages 2045–2053. Ieee, May 2007.
- [43] Wenbo He, Hoang Nguyen, Xue Liuy, Klara Nahrstedt, and Tarek Abdelzaher. iPDA: An Integrity-Protecting Private Data Aggregation Scheme for Wireless Sensor Networks. In *MILCOM 2008 - 2008 IEEE Military Communications Conference*, pages 1–7. Ieee, November 2008.
- [44] Wendi B Heinzelman, Amy L Murphy, Hervaldo S Carvalho, and Mark A Perillo. Middleware to support sensor network applications. *IEEE Network*, 18(1):6–14, 2004.
- [45] Christophe Huygens and Wouter Joosen. Federated and shared use of sensor networks through security middleware. In *Sixth International Conference on Information Technology: New Generations*, pages 1005–1011, 2009.
- [46] Christophe Huygens and Wouter Joosen. Federated and Shared Use of Sensor Networks through Security Middleware. *2009 Sixth International Conference on Information Technology: New Generations*, pages 1005–1011, April 2009.
- [47] P. Kamat, W. Trappe, and C. Ozturk. Enhancing Source-Location Privacy in Sensor Network Routing. In *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, pages 599–608. Ieee, 2005.
- [48] Kleomenis Katevas, Hamed Haddadi, and Laurissa Tokarchuk. Poster: Sensingkit: A multi-platform mobile sensing framework for large-scale experiments. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014.
- [49] Daniel Kifer and Ashwin Machanavajjhala. No free lunch in data privacy. In *Proceedings of the 2011 international conference on Management of data*, pages 193–204, 2011.

- [50] L. Krishnamachari, D. Estrin, and S. Wicker. The impact of data aggregation in wireless sensor networks. In *Proceedings 22nd International Conference on Distributed Computing Systems Workshops*, pages 575–578. IEEE Comput. Soc, 2002.
- [51] Balachander Krishnamurthy and Craig E. Wills. Privacy leakage in mobile online social networks. *Proceedings of the 3rd Wconference on Online social networks*, 2010.
- [52] Ilias Leontiadis, Christos Efstratiou, Cecilia Mascolo, and Jon Crowcroft. Sen-Share: Transforming Sensor Networks Into Multi-application Sensing Infrastructures. In *9th European Conference on Wireless Sensor Networks EWSN*, 2012.
- [53] Philip Levis and David Culler. Mate: a tiny virtual machine for sensor networks. *SIGOPS Oper. Syst. Rev.*, 36(5):85–95, 2002.
- [54] Kun Liu, Chris Giannella, and Hillol Kargupta. A Survey of Attack Techniques on Privacy-Preserving Data Perturbation. *Privacy-Preserving Data Mining: Models and Algorithms*, 2008.
- [55] T. Liu and Margaret Martonosi. Impala: a middleware system for managing autonomic, parallel sensor systems. In *ACM SIGPLAN Notices*, volume 38, pages 107–118. ACM, 2003.
- [56] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1(1):3–es, March 2007.
- [57] Alan Mainwaring, Joseph Polastre, Robert Szewczyk, David Culler, and John Anderson. Wireless Sensor Networks for Habitat Monitoring. In *International Workshop on Wireless Sensor Networks and Applications WSNA*. ACM, 2002.
- [58] Wassim Masri and Zoubir Mammeri. Middleware for Wireless Sensor Networks: A Comparative Analysis. *2007 IFIP International Conference on Network and Parallel Computing Workshops (NPC 2007)*, pages 349–356, September 2007.
- [59] Wassim Masri and Zoubir Mammeri. Middleware for Wireless Sensor Networks: Approaches, Challenges, and Projects. In *2007 IEEE International Conference on Signal Processing and Communications*, number November, pages 1399–1402. Ieee, November 2008.

- [60] Frank McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proceedings of the 35th SIGMOD international conference on Management of data*, pages 19–30. ACM, 2009.
- [61] Frank McSherry and Ratul Mahajan. Differentially-private network trace analysis. In *Proceedings of the ACM SIGCOMM 2010*, 2010.
- [62] Frank Mcsherry and Ilya Mironov. Differentially Private Recommender Systems: Building Privacy into the Netflix Prize Contenders. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 627–636, 2009.
- [63] Kiran Mehta, Donggang Liu, and Matthew Wright. Location Privacy in Sensor Networks Against a Global Eavesdropper. *2007 IEEE International Conference on Network Protocols*, pages 314–323, October 2007.
- [64] M.M. Molla and S.I. Ahamed. A Survey of Middleware for Sensor Network and Challenges. *2006 International Conference on Parallel Processing Workshops (ICPPW'06)*, pages 223–228, 2006.
- [65] Steven J. Murdoch. Hot or not: Revealing hidden services by their clock skew. In *13th ACM Conference on Computer and Communications Security*, pages 27–36. ACM Press, 2006.
- [66] Arvind Narayanan and Vitaly Shmatikov. Myths and fallacies of "personally identifiable information". *Communications of the ACM*, 53(6):24, June 2010.
- [67] Spiros Papadimitriou, Feifei Li, George Kollios, and Philip S. Yu. Time series compressibility and privacy. In *Proceedings of the 33rd international conference on Very large data bases*, pages 459–470. VLDB Endowment, 2007.
- [68] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5):51–58, 2000.
- [69] Vibhor Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *Proceedings of the 2010 international conference on Management of data*, pages 735–746. ACM, 2010.
- [70] Jason Reed, Adam J. Aviv, Daniel Wagner, Andreas Haeberlen, Benjamin C. Pierce, and Jonathan M. Smith. Differential privacy for collaborative security.

- In *Proceedings of the Third European Workshop on System Security - EUROSEC '10*, pages 1–7, New York, New York, USA, 2010. ACM Press.
- [71] Chao Ren, Xufei Mao, Ping Xu, GuoJun Dai, and ZhanHuai Li. Delay and energy efficiency tradeoffs for data collections and aggregation in large scale wireless sensor networks. In *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, pages 977–982. Ieee, October 2009.
- [72] Rathindra Sarathy and Krishnamurty Muralidhar. Evaluating Laplace Noise Addition to Satisfy Differential Privacy for Numeric Data. *Transactions on Data Privacy*, 4:1–17, 2011.
- [73] Bruce Schneier. *Applied Cryptography*. John Wiley & Sons, Inc., 1996.
- [74] Chien-Chung Shen, Chavalit Srisathapornphat, and Chaiporn Jaikaeo. Sensor information networking architecture and applications. *Personal communications, IEEE*, 8(4):52–59, 2001.
- [75] Eduardo Souto, Germano Guimarães, Glauco Vasconcelos, Mardoqueu Vieira, Nelson Rosa, and Carlos Ferraz. A message-oriented middleware for sensor networks. In *Proceedings of the 2nd Workshop on Middleware for Pervasive and Ad-hoc Computing*, pages 127–134. ACM, 2004.
- [76] Michal Sramka. A Privacy Attack That Removes the Majority of the Noise From Perturbed Data. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1–8. IEEE, 2010.
- [77] Vijay Srinivasan, John Stankovic, and Kamin Whitehouse. Protecting your daily in-home activity information from a wireless snooping attack. In *Proceedings of the 10th international conference on Ubiquitous computing - UbiComp '08*, pages 202–211, Seoul, Korea, 2008. ACM Press.
- [78] Mani Srivastava, Tarek Abdelzaher, and Boleslaw Szymanski. Human-centric sensing. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 370(1958):176–97, January 2012.
- [79] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty Fuzziness and*, 10(5):557 – 570, 2002.

- [80] Xueyan Tang and Jianliang Xu. Optimizing Lifetime for Continuous Data Aggregation With Precision Guarantees in Wireless Sensor Networks. *IEEE/ACM Transactions on Networking*, 16(4):904–917, August 2008.
- [81] Miao-Miao Wang, Jian-Nong Cao, Jing Li, and Sajal K. Dasi. Middleware for Wireless Sensor Networks: A Survey. *Journal of Computer Science and Technology*, 23(3):305–326, June 2008.
- [82] Hongkai Wen, Zhuoling Xiao, Andrew Symington, Andrew Markham, and Niki Trigoni. Comparison of accuracy estimation approaches for sensor networks. In *9th IEEE International Conference on Distributed Computing in Sensor Systems*, 2013.
- [83] Kui Wu, Dennis Dreef, Bo Sun, and Y. Xiao. Secure data aggregation without persistent cryptographic operations in wireless sensor networks. *Ad Hoc Networks*, 5(1):100–111, 2007.
- [84] J Yick, B Mukherjee, and D Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, August 2008.
- [85] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52:2292–2330, August 2008.
- [86] Wensheng Zhang, Chuang Wang, and Taiming Feng. GP²S: Generic Privacy-Preservation Solutions for Approximate Aggregation of Sensor Data (concise contribution). In *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 179–184. Ieee, March 2008.