PRIVACY IN EMERGING WIRELESS NETWORKS

by

PANDURANG KAMAT

A Dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Professor Wade Trappe

and approved by

____

____

____

____

New Brunswick, New Jersey

October, 2007

ABSTRACT OF THE DISSERTATION

Privacy in Emerging Wireless Networks

By PANDURANG KAMAT

Dissertation Director:
Professor Wade Trappe

Ad hoc wireless networks have emerged as a solution to providing ubiquitous, on-demand connectivity without the need for significant infrastructure deployment. In this thesis we address the privacy problems in two types of emerging wireless ad hoc networks, namely sensor and vehicular networks.

Although the content of sensor messages describing "events of interest" may be encrypted to provide confidentiality, the context surrounding these events may also be sensitive and therefore should be protected from eavesdroppers. The source-location privacy problem occurs in sensor networks when adversaries use RF localization techniques to perform hop-by-hop traceback of messages to the source sensor's location. Our work provides a formal model for this problem and examines the privacy characteristics of different sensor routing protocols. In order to provide efficient and private sensor communications, we devise new techniques to enhance source-location privacy that augment these routing protocols.

Similarly, an adversary armed with knowledge of the network deployment, routing algorithms, and the data sink location can infer the temporal patterns and track the spatio-temporal evolution of a sensed event, by monitoring the packet arrivals at the sink. We introduce the temporal privacy problem for delay-tolerant sensor networks, provide an information theoretic formulation and propose adaptive buffering to obfuscate temporal information from the adversary.

Sensor networks are also characterized by distinctive traffic patterns, wherein traffic mostly exists when events of interests occur. Due to the direct correlation between the type of event and size of data generated by it, an adversary observing a traffic burst can infer information about the type of event simply from the observed message size. We formulate this traffic privacy problem in terms of information entropy, present a quantifiable means to measure traffic privacy and propose solutions to enhance it.

Vehicular ad hoc networks represent a promising new communication paradigm that can facilitate many new forms of automotive applications. We present a robust and efficient security and privacy framework, for such networks, that uses identity-based cryptography. We show that our framework provides authentication, confidentiality, non-repudiation and message-integrity. Further, it supports scalable, user-customizable privacy through short-lived, unforgeable, pseudonyms.

Acknowledgements

This thesis is a major milestone in my educational journey that is, in many ways, more significant than the destination itself. Along the way many people have inspired, encouraged and guided me. I take this opportunity to express my sincere gratitude to them all.

First and foremost, I owe my most sincere thanks to my research advisor Dr. Wade Trappe, who has helped me become the researcher that I am today. From my early days as a student, meandering through dead-end research and stumbling over half-baked ideas, he prodded, cajoled, chastised and inspired me into a rigorous and confident researcher and helped me develop a keen analytical acumen. I thank him for his mentoring, his guidance and above all for his friendship. I also want to thank my co-advisor Dr. Yanyong Zhang, who has helped shape the systems side of my research. She has been the source of very valuable insight into the art and science of data interpretation, analysis and presentation. She has also been a good friend throughout and I have come to admire and emulate her strong work ethic.

During my tenure at WINLAB, I had the privilege of working with some amazing faculty members and colleagues. I learnt a lot from the vision and knowledge of Dr. Dipankar Raychaudhury and Ivan Seskar. As a member of the team that designed and developed the wireless networking testbed *ORBIT*, I was lucky to work with highly motivated and intelligent colleagues and friends like Sachin Ganu, Kishore Ramachandran, Zhibin Wu and Mesut Ali Ergin. Our seemingly endless strings of high-profile and high-pressure demos and presentations were made enjoyable by the camaraderie and commitment to excellence of my teammates.

Thanks are due to Joydeep Acharya and Suhas Mathur for being very valuable sounding boards, for their help in shaping the traffic privacy portion of my research and indulging me on my mathematical excursions. I also want to acknowledge my collaborators Wenyuan Xu and Rob Miller with whom I had the opportunity to engage in researching security in cognitive radio networks. I sincerely thank my thesis committee: Dr. Rich Martin, Dr. Naftaly Minsky and Dr. Siva Raj Rajagopalan (HP-Labs) for their feedback and valuable suggestions on my research. Thanks also to Dr. Stuart Haber of HP Labs for his mentoring during my internship.

I would like to thank Sridatt (James) Sugrim and Faiyaz Ahmed for the engineering support they provided to me and other students at WINLAB. Thanks also to Elaine Conners, Melissa Gelfman and Noreen DeCarlo who, as part of the administrative team at WINLAB, are the people behind the scenes, contributing to the smooth running and success of WINLAB and its students. I'm grateful to my dear friends Anand Mahale and Dr. Mahesh Ajgaonkar, who helped me acclimatize to life in the United States, exemplified discipline, compassion and humility for me and encouraged me to pursue a PhD. I'm also indebted to my extended family in the United States namely, Dr. Heramb, Dr. Madhumati, Nilkant and Dr. Asha Kamat for their constant support and encouragement.

This acknowledgement would be superfluous if I didn't acknowledge the very foundation of my being; my parents, Dr. Lata and Dr. Khivraj Kamat. They instilled in me the values of education, hard-work and humility at an early age. They gave me my faith and confidence and supported me in all my endeavors. My sister Tonima, who has been an ardent cheerleader for me, also deserves my thanks. Her love and admiration has always spurred me on to better myself.

Last, but certainly not the least, I wish to express my sincere gratitude to my wife Arati. She has been an endless source of inspiration and motivation throughout my graduate school journey. She has acted as a support structure for me at home while pursuing her own PhD and has shown me the way with her meticulousness, diligence and hard work. I'm lucky to have her by my side.

Dedication

To my parents

Dr. Lata and Dr. Khivraj Kamat

and

my wife

Arati

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction and Thesis Roadmap

If you don't know where you're going, you'll wind up somewhere else.[1]

Innovations in wireless networks are making them ubiquitous and fueling new applications that make use of the highly networked environment they provide. The open nature of the communication medium and the diversity of devices participating in the network create challenges in providing security and privacy in such networks. This thesis looks at privacy issues in two such emerging wireless networks, namely, wireless sensor networks and vehicular ad hoc networks.

1.1 What is Privacy ?

The Merriam-Webster dictionary defines privacy as *the quality or state of being apart from company or observation*. It is a fundamental, unalienable human right and the most primal of human desires. It is manifested in many forms, ranging from seclusion to secrecy to anonymity and pseudonymity. Totalitarian regimes around the world invade it with impunity while democratic governments tend to do so in their misguided efforts to provide security. Its meanings and implication vary widely. For most of us it is the ability to act in the confines of our home without prying eyes or the ability to walk the streets, unsurveiled or to not have our communication and reading habits snooped upon or our personal records accessed without authorization. The enemies of privacy, unlike the adversarial

[1] This quote and others that appear at the start of every chapter in this thesis are from Lawrence Peter ”Yogi” Berra, a famous baseball player and manager, also known for his humorous quotes called Yogiisms.

models in this thesis, often come armed not with antennas and packet snooping software but with National Security Letters and gag orders. In today's world, with the advent of technology and the ubiquitous presence of it, privacy is a scarce concept, getting scarcer. Scott McNealy, the former CEO of Sun Microsystems may well have been prophetic in proclaiming *"You have no privacy, get over it"*.

In computer and communication systems, privacy may be defined as the guarantee that information, in its general sense, is observable or decipherable by only those who are intentionally meant to observe or decipher it. The phrase "in its general sense" is meant to imply that there may be types of information besides data itself that are associated with the context surrounding the creation, storage and communication of that data.

Over the years, researchers have studied privacy issues in various sub-disciplines of Computer Science. Secure multi-party computation research has looked at how distinct multiple parties can jointly compute a function without learning any more information than is absolutely necessary to compute the function. Zero knowledge proofs and protocols have been developed for one party to prove the veracity of a mathematical statement to another without revealing anything but the statement. Privacy preserving databases and data-mining research has produced various techniques for data and query perturbation and clustering to protect the privacy of both the client and the server of the data within tolerable levels of errors in the results. Privacy research in communication networks has largely stemmed from Chaum's seminal work on Mixes and Mix networks, wherein he proposed a system of network nodes that would use encryption and buffering to de-correlate the sender and recipients of packets. High-profile products, such as the Mixmaster remailer and more recently the fully decentralized Tor anonymizing system, have emerged from subsequent research on this theme.

In this thesis we look at two specific types of emerging wireless networks and the privacy problems that crop up therein. We find that existing communication privacy techniques either fail to address or are incapable of tackling the privacy problems we identify in these networks due to the distinctive characteristics of both the networks and the problems themselves. We then go on to propose a suite of new techniques and frameworks that mitigate the problems and provide or enhance privacy in these domains.

A quick note before we go further: Just as in the case of designing a security system, a privacy system designer must follow Kerckhoffs' principle, which essentially states that the security of a system should only depend on the secrecy of a secret key and not that of any design aspect of the system including the algorithms. Guided by this principle, in all our system and algorithm designs we assume powerful adversary models with full knowledge of the system design, nature of the applications and routing and privacy algorithms in use.

1.2 Wireless sensor networks

The coupling between advancements in wireless communication technologies and low-cost hardware has initiated a change in the purpose for which networks are used. Increasingly affordable sensors are being developed that can operate for long periods of time without requiring external power, and can gather a broad spectrum of data such as temperature measurements in office buildings, pollution readings in ecologically sensitive environments, or cardiologic data for at-risk heart patients. Simultaneously, there have been significant developments in communication technologies, such as low-power techniques capable of reliably transmitting information between radio devices, and new networking paradigms that will allow radio nodes to form ad hoc relationships with each other that can adjust to changing radio environments. Together, these advancements will support the deployment of networks capable of supplying vast quantities of strategic and timely data, which will facilitate new classes of remote-sensing and monitoring applications. Sensor networks also promise to have a significant commercial impact, with the emerging market for wireless sensor devices estimated to reach over $10B in 2010 [1] and comparable figures projected for the corresponding software application markets. It is therefore evident that sensor-driven applications are expected to have a huge impact on our society. *However, in spite of the predicted demand for sensor applications, there are several disruptive challenges lurking in the future that threaten the successful deployment of sensor networks and how smoothly they will be integrated into our daily lives. Perhaps core amongst these challenges are issues associated with security and privacy.*

Security and privacy for sensor networks is complicated by the fact that sensor networks will be commodity networks, consisting of low-cost devices that will employ readily available wireless communication technologies. As an example, Berkeley Motes employ a tunable radio technology that is easily observable by spectrum analyzers, while other examples exist of sensor devices employing low-power versions of 802.11 wireless technologies. Consequently, adversaries will be able to easily gain access to communications between sensor nodes either by purchasing their own low-cost sensor device and running it in a monitor mode, or by employing slightly more sophisticated software radios capable of monitoring a broad array of radio technologies [2]. Further, adversaries might be able to compromise a small, strategic set of sensor nodes, and use these to mount a variety of security attacks, such as injecting false sensor data into the network, launching denial of service attacks against the data sink or other sensor nodes, or even disrupting the routing and delivery of legitimate sensor data.

Overall, we may categorize the types of security and privacy threats into two broad classes: content-oriented security/privacy threats, and contextual privacy threats. Content-oriented security and privacy threats are issues that arise due to the ability of the adversary to observe and manipulate the *exact* content of packets being sent over the sensor network, whether these packets correspond to actual sensed-data or sensitive lower-layer control information. The issue of contextual privacy is more elusive to define for sensor networks than content-oriented security. Essentially, it is concerned with protecting the *context* surrounding the measurement and transmission of sensed data. The context can consist of many aspects, some of which are temporal, source and destination location, source and destination identity, and traffic size. There may be other contextual aspects that might arise, and one challenge with contextual privacy is that it has various dimensions and target issues that are not always a priori identifiable.

Although issues related to sensor security are important, we believe many of the core problems associated with sensor security are on the road to eventual resolution due to an abundance of recent research by the technical community, c.f. [3–5]. Contextual privacy issues associated with sensor communication, however, have not been as thoroughly addressed. In this thesis we develop solutions for following contextual privacy challenges in sensor network communications:

- **Source Location Privacy:** The physical or virtual location of communication participants may be sensitive information that is undesirable for an adversary to know. This thesis provides a formal model for the source-location privacy problem in sensor networks and examines the privacy characteristics of different sensor routing protocols. In order to provide efficient and private sensor communications, we devised new techniques to enhance source-location privacy that augment these routing protocols.

- **Temporal Privacy:** Paired with the location of the data source is the time at which the data was created. Together, the availability of spatial and temporal information to an adversary constitute a serious privacy breach as this information allows an adversary to track the information origin. We define this temporal privacy problem for delay-tolerant sensor networks, provide an information theoretic formulation and propose adaptive buffering at intermediate nodes to obfuscate temporal information from the adversary.

- **Traffic Privacy:** The size of a messages in a sensor network can allow an adversary to infer certain information. Although an adversary might not be able to decrypt sensor messages, by observing the size of the packets and the amount of traffic crossing the sensor network, he might be able to deduce information about the situation in which the data was generated. This thesis formulates the traffic privacy problem in terms of information entropy and proposes various message padding solutions to alleviate it. We then present a quantifiable means to measure the traffic privacy of a network and present an algorithm to maximize it.

Addressing privacy issues for sensor networks is very challenging, and requires strategies that differ from those for traditional networks [6–8]. In particular, sensor devices will be low-powered, and therefore will not have the same amount of resources available as their wired counterparts. Thus resource-efficiency and other important systems-level issues must be examined while developing privacy mechanisms. Additionally, since the theoretical attacks of today are often the realizable attacks of tomorrow, it is important to construct theoretical attack models that facilitate the investigation of mature privacy-preserving countermeasures. This thesis takes a coordinated *theory-meets-systems* approach by first developing theoretical formulations for privacy and then by evaluating the proposed privacy enhancements using network simulations.

1.3 Vehicular ad hoc networks

As on-board computation and communication capabilities of vehicles improve, we continue to approach a future where cars can conduct full-scale communication with roadside infrastructure as well as with other automobiles. Vehicular ad hoc networks (VANETs) represent a promising new communication technology that can facilitate many new forms of automotive applications. Many of the applications that will run on these networks will require a high degree of security and privacy. Researchers are exploring the feasibility of vehicular applications, ranging from enhancing driver safety to traffic management to providing roadside services and infotainment using inter-vehicle communications. Unfortunately, as has occurred for other types of networks, the security of this new communication modality has largely been considered an afterthought. However, security is especially critical in vehicular communications. A less than perfect communication system can endanger people's lives and can be responsible for more accidents and abuse of the system. Privacy is equally important, as drivers will want the assurance that they cannot be tracked inappropriately, for example by other drivers or by law enforcement agencies without proper authorization. This poses several interesting challenges in designing services for vehicular networks. In this thesis we address the challenge of designing a comprehensive security and privacy framework for vehicular ad hoc networks that provides secure and pseudonymous communication using identity-based cryptographic constructs.

1.4 Thesis roadmap

This thesis is divided into two broad sections. The next three chapters of this thesis explore the contextual privacy issues in wireless sensor networks and Chapter 5 then looks at the privacy challenges in vehicular ad hoc networks. The chapters are organized as follows:

In Chapter 2, we identify the problem of source-location privacy in sensor networks. Specifically we study the challenge of protecting the location of the source of a message stream from an adversary who may use RF localization techniques to perform hop-by-hop traceback of messages to the source sensor's location. Our work provides a formal model for the source-location privacy problem in sensor networks and examines the privacy characteristics of different sensor routing protocols. We examine two popular classes of routing protocols: the class of flooding protocols, and the class of single path routing protocols. We find that most of the current protocols cannot provide efficient source-location privacy while maintaining desirable system performance. In order to provide efficient and private sensor communications, we devised new techniques to enhance source-location privacy that augment these routing protocols. One of our strategies, a technique we have called phantom routing, has proven flexible and capable of protecting the source's location, while not incurring a noticeable increase in energy overhead. We also examine the effect of node density and source mobility on location privacy. Further, we demonstrate the efficacy of phantom routing against different adversary strategies and capabilities including a distributed adversary.

In Chapter 3, we introduce the problem of temporal privacy in delay-tolerant sensor networks. We show how an adversary armed with knowledge of the network deployment, routing algorithms, and the base-station (data sink) location can infer the temporal patterns of interesting events by merely monitoring the arrival of packets at the sink, thereby allowing the adversary to remotely track the spatio-temporal evolution of a sensed event. We propose the use of adaptive buffering at intermediate nodes on the source-sink routing path to obfuscate temporal information from the adversary. We first present the effect of buffering on temporal privacy using an information-theoretic formulation, and then examine the effect that delaying packets has on buffer occupancy. We observe that temporal privacy and efficient buffer utilization are contrary objectives, and then present an adaptive buffering strategy that effectively manages these tradeoffs. Finally, we evaluate our privacy enhancement strategies using simulations, where privacy is quantified in terms of the adversary's mean square error.

Sensor networks are also characterized by distinctive traffic patterns, wherein traffic mostly exists when events of interests occur. Data relayed for each type of event dictates the size of the message sent by a sensor node. In a multi-application sensor network, an adversary observing a traffic burst can infer information about the type of event simply from the message size of the transmission he intercepts. In Chapter 4, we formulate this traffic privacy problem in terms of information entropy and propose various message padding solutions to alleviate it. We present a quantifiable means to measure the traffic privacy of a network and present an algorithm to maximize it given a a bit budget to spend on message padding.

Chapter 5 of the thesis focuses on vehicular ad hoc networks. In this chapter, we present a robust and efficient security and privacy framework for vehicular networks suited for both inter-vehicular and vehicle-to-infrastructure communication. Our system uses identity-based cryptography to provide authentication, confidentiality, non-repudiation and message-integrity. Additionally it provides scalable, user-customizable privacy using short-lived, authenticated and unforgeable, pseudonyms. This feature can be used by VANET applications that require quantifiable trust and privacy to provide differentiated service based on various levels of trust and privacy thresholds.

Finally, in Chapter 6 we conclude the thesis and identify directions for future research.

Chapter 2

Location Privacy in Wireless Sensor Networks

When you come to the fork in the road, take it.

One important class of future sensor-driven applications will be applications that monitor a valuable asset. For example, sensors will be deployed in natural habitats to monitor endangered animals, or may be used in tactical military deployments to provide information to networked operations. In these asset monitoring applications, it is important to provide confidentiality to the source sensor's location.

Many of the privacy techniques employed in general network scenarios are not appropriate for protecting the source location in a sensor network [6, 9–11]. This is partially due to the fact that the problems are different, and partially due to the fact that many of the methods introduce overhead which is too burdensome for sensor networks. One notable challenge that arises in sensor networks is that the shared wireless medium makes it feasible for an adversary to locate the origin of a radio transmission, thereby facilitating hop-by-hop traceback to the origin of a multi-hop communication.

To address source-location privacy for sensor networks, this thesis provides a formal model for the source-location privacy problem and examines the privacy characteristics of different sensor routing protocols. We introduce two metrics for quantifying source-location privacy in sensor networks, the safety period and capture likelihood. In our examination of popular routing techniques used in today's sensor networks, we also considered important systems issues, like energy consumption, and found that most protocols cannot provide efficient source-location privacy. We propose new techniques to enhance source-location privacy that augment these routing protocols. It is important that this privacy enhancement does not come at a cost of a significant increase in resource consumption. We have devised a strategy, called phantom routing, that has proven flexible and capable of preventing the adversary from tracking the source location with minimal increase in energy overhead.

In Section 3.5 we describe the problem domain in detail including the network and adversary model. In Section 2.3 we discuss baseline routing techniques and their performance in terms of privacy. The feasibility of using fake message injection to enhance privacy is studied in Subsection 2.3.2. Phantom routing technique to enhance privacy of routing algorithms is introduced in Subsection 2.3.3. Performance of phantom routing under different network conditions and adversarial models is analyzed in Section 3.5.3 and 2.5 respectively.

2.1 Asset Monitoring Sensor Networks

In order to facilitate the discussion and analysis of source-location privacy in sensor networks, we need to select an exemplary scenario that captures most of the relevant features of both sensor networks and potential adversaries in asset monitoring applications. Throughout this chapter, we use a generic asset monitoring application, which we have called the *Panda-Hunter Game*, as well as refer to a formal model for asset monitoring applications that can benefit from source-location privacy protection. In this section we begin by introducing the Panda-Hunter Game and the formal model, and then discuss how to model the Panda-Hunter Game using a discrete, event-driven simulation framework.

In the Panda-Hunter Game, a large array of panda-detection sensor nodes have been deployed by the Save-The-Panda Organization to monitor a vast habitat for pandas [12]. As soon as a panda is observed, the corresponding *source* node will make observations, and report data periodically to the *sink* via multi-hop routing techniques. The game also features a hunter in the role of the adversary, who tries to capture the panda by back-tracing the routing path until it reaches the source. As a result, a privacy-cautious routing technique should prevent the hunter from locating the source, while delivering the data to the sink.

In the Panda-Hunter Game, we assume there is only a single panda, thus *a single source*, and this source can be either stationary or mobile. During the lifetime of the network, the sensor nodes will continually send data, and the hunter may use this to his advantage to track and hunt the panda. We assume that the source includes its ID in the encrypted messages, but only the sink can tell a node's location from its ID. As a result, even if the hunter is able to break the encryption in a reasonably short time frame, it cannot tell the source's location. In addition, the hunter has the following characteristics:

- **Non-malicious:** The hunter does not interfere with the proper functioning of the network, otherwise intrusion detection measures might flag the hunter's presence. For example, the hunter does not modify packets in transit, alter the routing path, or destroy sensor devices.

- **Device-rich:** The hunter is equipped with devices, such as antenna and spectrum analyzers, so that it can measure the angle of arrival of a message and the received signal strength. From these two measurements, after it hears a message, it is able to identify the immediate sender and move to that node. We emphasize, though, that the hunter cannot learn the origin of a message packet by merely observing a relayed version of a packet. In addition, the hunter can detect the panda when it is near.

- **Resource-rich:** The hunter can move at any rate and has an unlimited amount of power. In addition, it also has a large amount of memory to keep track of information such as messages that have been heard and nodes that have been visited.

- **Informed:** To appropriately study privacy, we must apply Kerckhoff's Principle from security to the privacy setting [13]. In particular, Kerckhoff's Principle states that, in assessing the privacy of a system, one should always assume that the enemy knows the methods being used by the system. Therefore, we assume that the hunter knows the location of the sink node and knows various methods being used by the sensor network to protect the panda.

2.1.1 A Formal Model

In order to understand the issue of location privacy in sensor communication, we now provide a formal model for the privacy problem. Our formal model involves the definition of a general asset monitoring network game, which contains the features of the Panda-Hunter game analyzed in this chapter.

Definition 1 *An asset monitoring network game is a six-tuple* $(\mathcal{N}, S, A, \mathcal{R}, \mathcal{H}, \mathcal{M})$*, where*

1. $\mathcal{N} = \{n_i\}_{i \in I}$ *is the network of sensor nodes* n_i*, which are indexed using an index set* I*.*

2. S *is the network sink, to which all communication in the sensor network must ultimately be routed to.*

3. A *is an asset that the sensor network monitors. Assets are characterized by the mobility pattern that they follow.*

4. $\mathcal{R}$ *is the routing policy employed by the sensors to protect the asset from being acquired or tracked by the hunter* $\mathcal{H}$*.*

5. $\mathcal{H}$ *is the hunter, or adversary, who seeks to acquire or capture the asset* A *through a set of movement rules* $\mathcal{M}$*.*

The game progresses in time with the sensor node that is monitoring the asset periodically sending out messages.

The purpose of the network is to monitor the asset, while the purpose of the routing strategy is two-fold: to deliver messages to the sink and to enhance the location-privacy of the asset in the presence of an adversarial hunter following a movement strategy. We are therefore interested in privacy measures and network efficiency metrics.

Definition 2 *The privacy associated with a sensor network's routing strategy $\mathcal{R}$ can be quantified through two differing performance metrics:*

1. *The safety period Φ of a routing protocol $\mathcal{R}$ for a given adversarial movement strategy $\mathcal{M}$ is the number of new messages initiated by the source node that is monitoring an asset, before the adversary locates the asset.*
2. *The capture likelihood L of a routing protocol $\mathcal{R}$ for a given adversarial movement strategy $\mathcal{M}$ is the probability that the adversary can capture the asset within a specified time period.*

On the other hand, the network's performance may be quantified in terms of its energy consumption, and the delivery quality. A sensor node consumes energy when it is sending messages, receiving messages, idling, computing, or sensing the physical world. Among all the operations, sending and receiving messages consume the most energy [14, 15]. We measure the energy consumed in a sensor network by the total number of messages that are sent by all the nodes within the entire network until the asset is captured. We assume that messages are all the same length, each sensor transmits with the same transmission power, and hence each transmission by each sensor requires an equal amount of energy. Consequently, the greater the amount of messages required by a strategy, the more energy that strategy consumes. We use two metrics to measure the delivery quality. One is the average message latency, and the other is the event delivery ratio.

In order to illustrate the formal model of the asset monitoring game, we examine a special case of the Panda-Hunter Game. Suppose that we have a sensor network $\mathcal{N} = \{n_i\}$, where nodes n_i are located on a two-dimensional integer grid and that one of these nodes is designated as the network sink. Network devices might monitor a stationary panda, i.e. the asset A, located at a particular sensing node n_A. This node will periodically transmit sensor messages to the sink S following a routing policy $\mathcal{R}$. One possible routing policy $\mathcal{R}$ might be to employ shortest-path routing in which a single route is formed between the source and sink S according to a gradient-based approach. A hunter $\mathcal{H}$, might start at the network sink S, and might follow a movement strategy $\mathcal{M}$. One possible movement strategy could involve $\mathcal{H}$ repeatedly determining the position of the node that relayed the sensor message and moving to that relay node. Another movement strategy might involve $\mathcal{H}$ initially moving two hops, in order to get a head start, and then continue by moving one hop at a time. The safety period Φ corresponds to the amount of messages transmitted by the source which, in the case of the first movement strategy, corresponds directly to the amount of time it takes the hunter to reach the panda. On the other hand, there is a possibility, in the second movement strategy, that the hunter might skip past the panda (when the panda is one hop from the sink), in which case the hunter will miss the panda entirely and thus $L \neq 1$. Clearly, both the safety period Φ and the capture likelihood L depend on the location of the panda, the mobility of the panda, the routing strategy $\mathcal{R}$ and the movement rules $\mathcal{M}$ for the hunter.

2.2 Simulation Model

We have built a discrete event-based simulator to study the privacy protection of several routing techniques. We are particularly interested in large-scale sensor networks where there is a reasonably large separation between the source and the sink. In order to support a large number of nodes in our simulations, we have made a few approximations. Unless otherwise noted, for the simulation results provided in this chapter, we have a network $\mathcal{N}$ of 10,000 randomly located nodes, and the hunter had a hearing radius equal to the sensor transmission radius.

In reality, wireless communication within one hop involves channel sensing (including backoffs) and MAC-layer retransmissions due to collisions. Our simulator ignores the collisions. We emphasize that this should not have a noticeable effect on our accuracy for the following reasons. First, when more reliable MAC protocols are employed, the probability of collision decreases considerably, and channel sensing time may go up correspondingly. Second, sensor networks usually involve light traffic loads with small packets, which result in a lower likelihood of collisions. As a result, our simulator focuses on the channel sensing part. We employ a simple channel sensing model: if a node has m neighbors that may send packets concurrently, the gap before its transmission is a uniformly distributed random number between 1 and m clock ticks. Further, we argue that, although the absolute numbers we report in this work may not directly calibrate to a real network, the observed performance trends should hold.

Another approximation is the way we keep track of energy consumption. Earlier studies have pointed out that the bulk of a sensor node's energy is spent by its radio. Specifically, once the radio is on, no matter what state it is in, e.g., transmission, reception, channel sensing, or idling, the power consumption rate is on the same magnitude. Routing techniques, however, only directly affect the number of messages being exchanged, and thus we use this to measure the energy consumption. Further, due to the nature of wireless media, the numbers of transmissions and receptions are proportional, the latter being m times of the former, wherein m is the average number of neighbors a node has. As a result, we use the number of transmissions alone to measure the energy consumption of routing protocols.

Next, let us look at how we implement the Panda-Hunter game in our simulator. In the game, the panda pops up at a random location. Section 2.3 considers the scenario where the panda stays at the source until it is caught, while Section 2.4.2 investigates how the routing techniques perform for a moving panda. Once the hunter gets close to the panda (i.e., within Δ hops from the panda), the panda is considered captured and the game is over. As soon as the panda appears at a location, the closest sensor node, which becomes the source, will start sending packets to the sink reporting its observations. The simulator uses a global clock and a global event queue to schedule all the activities within the network, including message sends, receives and data collections. The source generates a new packet every T clock ticks until the simulation ends, which occurs either when the hunter catches the panda or when the hunter cannot catch the panda within a threshold amount of time (e.g. the panda has returned to its cave).

2.3 Privacy Protection for a Stationary Source

Rather than build a completely new layer for privacy, we take the viewpoint that existing technologies can be suitably modified to achieve desirable levels of privacy. We will therefore examine several existing routing schemes $\mathcal{R}$ to protect the source's location, while simultaneously exploring how much energy they consume. Specifically, we explore two popular classes of routing mechanisms for sensor networks: flooding and single-path routing. For each of these techniques, we propose modifications that allow for enhanced preservation of the source's location or allow us to achieve improved energy conservation. After exploring each of these two classes, we combine our observations to propose a new technique, which we call *phantom routing*, which has both a flooding and single-path variation. Phantom routing is a powerful and effective privacy enhancing strategy that carefully balances the tradeoffs between privacy and energy consumption.

2.3.1 Baseline Routing Techniques

In sensor networks, flooding-based routing and single-path routing are the two most popular classes of routing techniques. In this study, we first examine baseline routing strategies $\mathcal{R}$ from these two classes, and examine their capabilities in protecting the source-location privacy as well as in conserving energy in great depth.

Flooding-based Routing

Many sensor networks employ flooding to disseminate data and control messages [16–19]. In flooding, a message originator transmits its message to each of its neighbors, who in turn retransmit the message to each of their neighbors. Although flooding is known to have performance drawbacks, it nonetheless remains a popular technique for relaying information due to its ease of implementation, and the fact that minor modifications allow it to perform relatively well [20, 21].

In our baseline implementation of flooding, we have ensured that every node in the network only forwards a message once, and no node retransmits a message that it has previously transmitted. When a message reaches an intermediate node, the node first checks whether it has received that message before. If this is its first time, the node will broadcast the message to all its neighbors. Otherwise, it just discards the message. Realistically, this would require a cache at each sensor node. However, the cache size can be easily kept very small because we only need to store the sequence number of each message. We assume that each intermediate sensor node can successfully decrypt just the portion of the message corresponding to the sequence number to obtain the sequence number. Such an operation can easily be done using the CTR-mode of encryption. It is thus reasonable to expect that each sensor device will have enough cache to keep track of enough messages to determine whether it has seen a message before.

Probabilistic flooding [20, 21] was first proposed as an optimization of the baseline flooding technique to cut down energy consumption. In probabilistic flooding, only a subset of nodes within the entire network participate in data forwarding, while the others simply discard the messages they receive. The probability that a node forwards a message is referred to as the *forwarding probability* ($P_{forward}$), and plain flooding can be viewed as probabilistic flooding with $P_{forward} = 1$.

In our simulation, we implement probabilistic flooding as follows. Every time a node receives a new message (it discards the message that it has received before no matter whether it has forwarded it or not), it generates a random number q that is uniformly distributed between 0 and 1. If $q < P_{forward}$, the node will forward/broadcast this message to its neighbors. Otherwise, it will just discard that message. The parameter, $P_{forward}$, is important to the overall performance of this approach. A small value can help reduce the energy consumption though at the expense of lower network coverage and connectivity, while a large value can ensure a higher network coverage and connectivity but will have a correspondingly higher energy consumption.

Single-Path Routing

Unlike flooding, a large number of energy-efficient routing techniques allow a node to forward packets only to one of (or a small subset of) its neighbors. This family of routing techniques is referred to as *single-path routing* in our work (e.g., GPSR [22], trajectory-based routing [23], directed diffusion [18], etc). Single-path routing techniques usually require either extra hardware support or a pre-configuration phase. For example, in [22], Karp and Kung propose to use the location information of a node, its neighbors and the destination to calculate a greedy single routing path. In [23], Niculescu and Nath propose trajectory-based routing, which uses the location information associated with a node and its neighbors to create a routing path along a specified trajectory. Such location information can be obtained by either using GPS or other means. In Directed Diffusion [18], an initial phase sets up the “gradients” from each sensor node towards the sink. Later in the routing phase, each intermediate forwarding node can use its neighbors' gradients to implement single-path routing. Whenever the source or the sink changes, a re-configuration stage is required in order to reset the routes.

In this study, we try not to assume extra hardware for a normal sensor node. Instead, we use an initial configuration phase to set up the gradients, i.e. hop count between each node and the sink. In the configuration phase, the sink initiates a flood, setting the initial hop count to 0. Any intermediate node will receive the packet many times. It makes sure that it only processes the packet from all of its neighbors once, discarding duplicates. Every time it receives the message, it increments the hop in the message, records it in its local memory, and then broadcasts to its neighbors. After the initial phase, among all the hop counts it has recorded, a sensor node chooses the minimum value as the number of hops from the sink, and updates its neighbors with that number. Then, every sensor node maintains a neighbor list, which is rank-sorted in ascending order according to each neighbor's hop count to the sink. The head of the list, which has the shortest distance to the sink, is said to have the maximum gradient towards the sink. In the baseline single-path routing protocol, as soon as the source generates a new packet, it forwards the packet to the neighbor with the maximum gradient. Every node along the routing path will repeat this process until the packet reaches the sink. Our version of single-path routing thus corresponds to shortest-path routing, and we use these two terms interchangeably.

Adversary Model and Performance Comparison

Before we delve into the location-privacy protection capability of routing techniques, we define one class of hunter $\mathcal{H}$. In Algorithm 1, the hunter follows a simple but natural adversary model, where the adversary starts from the sink, waits at a location until it hears a new message, and then moves to the immediate sender of that message. It repeats this sequence until it reaches the source

```
Algorithm: Adversary Strategy I: Patient Adversary $\mathcal{H}$
next_location = sink;
while (next_location != source) do
    Listen(next_location);
    msg = ReceiveMessage();
    if (IsNewMessage(msg)) then
        next_location = CalculateImmediateSender(msg);
        MoveTo(next_location);
    end
end
```

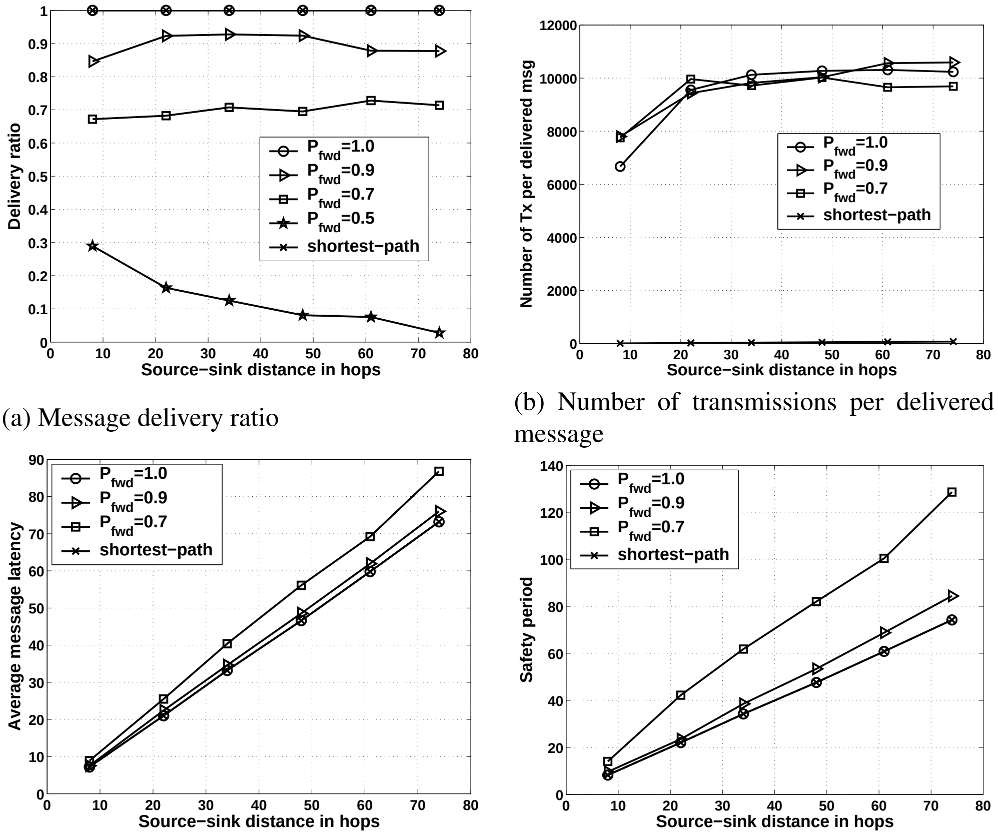
Algorithm 1: *The adversary waits at a location until it receives a new message.*

location. In this model, the adversary assumes that as long as he is patient enough, he will obtain some information that can direct him to the source. We thus refer to this $\mathcal{H}$ model as a *patient adversary*.

Figures 5.2(a)-(d) provide the performance of these baseline routing techniques for a patient adversary for different source-sink distances. In this set of results, we have 10,000 nodes uniformly randomly distributed over a 6000 × 6000 (m^2) network field. The average number of neighbors is 8.5. Among 10,000 nodes, less than 1% are weakly connected with less than 3 neighbors.

Delivery Quality

As expected, baseline flooding and shortest-path routing both give good delivery quality, namely, 100% delivery ratio (Figure 5.2(a)) and lowest message latency (Figure 5.2(c)). On the other hand, probabilistic flooding may have a poorer delivery quality. In particular, we find that probabilistic flooding techniques with $P_{forward} < 0.7$ result in a low message delivery ratio, especially when the source and the sink are far apart. Figure 5.2(a) shows that for $P_{forward} = 0.5$, the message delivery ratio can drop below 5%. As a result, we focus our attention on probabilistic flooding techniques with $P_{forward} \geq 0.7$ in the discussion below.



(a) Message delivery ratio

(b) Number of transmissions per delivered message

(c) Message latency

(d) Safety period

Figure 2.1: Performance of baseline routing techniques.

Energy Consumption

We use the number of transmissions to measure energy consumption, and instead of using the total energy consumed, we report energy consumption per successfully delivered message since some of the messages may not reach the sink (for probabilistic flooding) and this metric captures the wasted energy. For baseline flooding, every message can successfully reach the sink, and each message incurs n transmissions, where n is the number of sensor nodes in the network. Similarly, single-path routing can deliver all the messages, while each message incurs h transmissions where h is the number of hops in the shortest source-sink path. The number of transmissions per successfully delivered message is more complicated for probabilistic flooding schemes. Each successfully delivered message incurs $nP_{forward}$ transmissions, yet there is no guarantee that each message reaches the sink. This behavior has been studied thoroughly by the community [20, 21].

The effective energy usage is reported in Figure 5.2(b). Shortest-path routing incurs a much lower energy consumption (h as we discussed above). Three flooding-based techniques have similar energy consumption figures for each successfully delivered message (n as we discussed above). We would like to point out that those data points below $n = 10,000$ for nearby source-sink configurations are because we stopped the simulation as soon as the panda was caught and the flooding of messages had not yet finished.

Privacy Protection

Although single-path protocols have desirable energy consumption since they reduce the number of messages sent/received, they are rather poor at protecting the source location privacy (Figure 5.2(d)). Since only the nodes that are on the routing path forward messages, the adversary can track the path easily, and can locate the source within h moves. The safety period Φ of baseline single-path routing protocols is the same as the length of the shortest routing path because the adversary can observe every single message the source transmits.

At first glance, one may think that flooding can provide strong privacy protection since almost every node in the network will participate in data forwarding, and that the adversary may be led to the wrong source. Further inspection, however, reveals the contrary. We would like to emphasize that *flooding provides the least possible privacy protection as it allows the adversary to track and reach the source location within the minimum safety period.* Figure 5.2(d) shows that flooding and shortest-path routing lead to the same minimal privacy level. Specifically, the safety period is the same as the hop count on the shortest path.

The poor privacy performance of flooding can be explained by considering the set of all paths produced by the flooding of a single message. This set consists of a mixture of different paths. In particular, this set contains the shortest source-sink path. The shortest path is more likely to reach the hunter first, and thus the hunter will always select the shortest path out of all paths produced by flooding.

In addition to its energy efficiency, probabilistic flooding can improve the privacy protection as well. Imagine there exists a path $\{1, 2, 3, 4, sink\}$, and the adversary is waiting for a new message at node 4. In flooding, the subsequent message will certainly arrive at node 4. However, in probabilistic flooding, the subsequent message may not arrive at node 4 because neighboring nodes may not forward, or take longer to arrive. As a result, the source will likely have to transmit more messages in order for the adversary to work his way back to the source. The more messages the adversary misses, the larger the safety period for the panda, and hence source location protection is provided.

The primary observation is that it is hard for probabilistic flooding techniques to strike a good balance between privacy protection and delivery ratio. For instance, in our study, probabilistic flooding with $P_{forward} = 0.7$ can improve the safety period of baseline flooding roughly by a factor of 2. At the same time, however, it has a message delivery ratio of 70%, which may not be enough for some applications. On the other hand, $P_{forward} = 0.9$ can give a good delivery ratio, but its privacy level is only marginally improved compared to baseline flooding.

2.3.2 Routing with Fake Sources

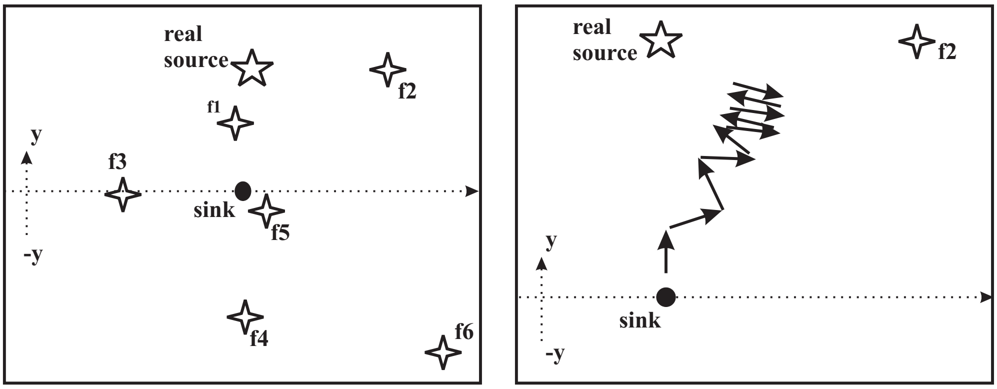
Baseline flooding and single-path routing cannot provide privacy protection because the adversary can easily identify the shortest path between the source and the sink. This behavior may be considered a result of the fact that there is a single source in the network, and that messaging naturally pulls the hunter to the source. This suggests that one approach we can take to alleviate the risk of a source-location privacy breach is to devise new routing protocols $\mathcal{R}$ that introduce more sources that inject fake messages into the network.

In order to demonstrate the effectiveness of fake messaging, we assume that these messages are of the same length as the real messages, and that they are encrypted as well. Therefore, the adversary cannot tell the difference between a fake message and a real one. As a result, when a fake message reaches the hunter, he will think that it is a legitimate new message, and will be guided towards the fake source.

One challenge with this approach is how to inject fake messages. We need to first decide how to create the fake sources, and when and how often these fake sources should inject false messages. Specifically, we want these fake sources to start only after the event is observed, otherwise the use of fake sources would consume precious sensor energy although there is no panda present to protect.

First, let us look at one naive injection strategy that does not require any additional overhead, which we refer to as the *Short-lived Fake Source* routing strategy. This strategy uses the constant P_{fake} to govern the fake message rate, and choose $P_{fake} \propto \frac{1}{n}$. For any node within the network, after it receives a real message, it generates a random number q that is uniformly distributed between 0 and 1. If $q < P_{fake}$, then this node will produce a fake packet and flood it to the network. In this strategy, the fake source changes from one fake message to another. Although this strategy is easy to implement, it does not improve the privacy level of baseline flooding because the fake sources are short-lived. Even if the hunter is guided by one fake message towards a wrong location, there are no subsequent fake messages around that location to draw him even further away, so he can catch the next real message. As a result, we need a persistent fake source to mislead the hunter.

Thus, we introduce a *Persistent Fake Source* routing strategy. The basic idea of this method is that once a node decides to become a fake source, it will keep generating fake messages regularly so that the hunter can be misled. It is intuitive that a fake source close to the real source, or on the way



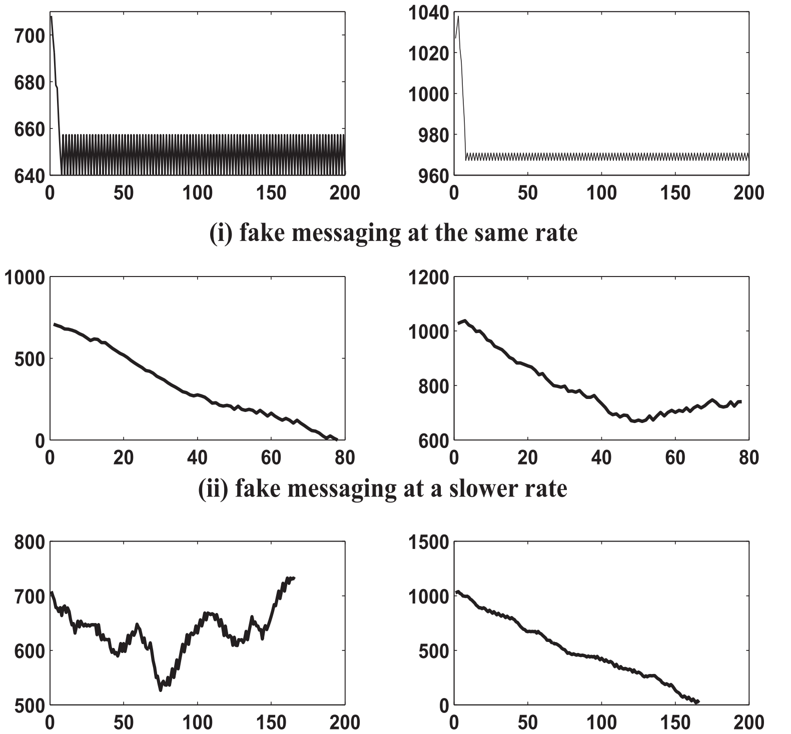
(a) Different locations of fake sources

(b) Pulls from both real source and fake source

Figure 2.2: Routing with fake sources.

from the sink to the source, can only help lead the adversary towards the real source, thus providing a poor privacy protection (such as $f1$ in Figure 2.2(a)). As a result, locations $f2, f3, f4, f5, f6$ are better alternatives in terms of protecting privacy. Among these locations, we would like to point out that the distances of the fake sources to the sink should be considered as well when choosing a fake source. For example, if a fake source is too far away from the sink compared to the real source, such as $f6$ in our example, then it would not be as effective in pulling the adversary. On the other hand, if a fake source is too close to the sink, it can draw the hunter quickly towards its location, and as we mention below, a hunter can easily detect the fake source in such cases. As a result, we conclude that the fake sources should be comparable to the real source with respect to their distances to the sink. Hence, $f2, f3$, and $f4$ are good candidates.

The above discussion assumes that we have the global picture of the network deployment. There are many ways of implementing this in a distributed manner, and in this study, we discuss a simple way where we assume that each node knows the hop count between itself and the sink, and that the sink has a sectional antenna. The first assumption can be achieved by a simple flood from the sink, as described in Section 2.3.1. The second assumption is valid because sinks usually are much more



(i) fake messaging at the same rate

(ii) fake messaging at a slower rate

(iii) fake messaging at a faster rate

(a) Time series of the hunter's distance from the real source

(b) Time series of the hunter's distance from the fake source

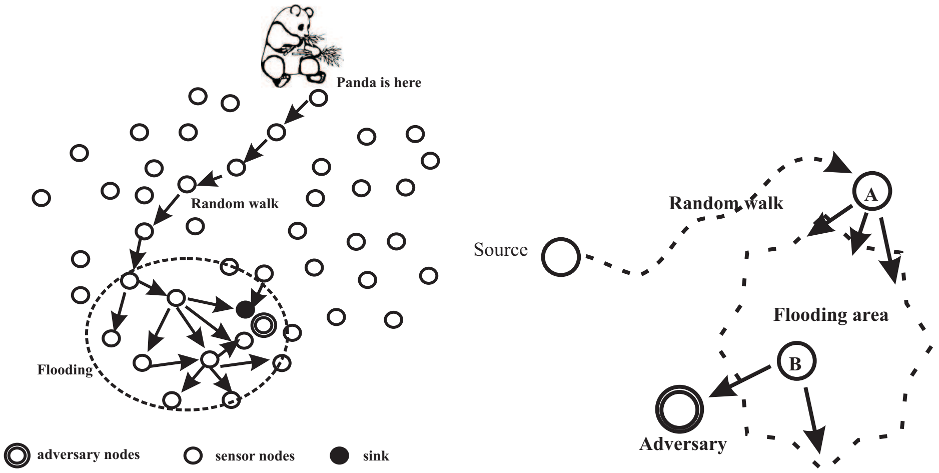
Figure 2.3: Fake messaging rates.

powerful than normal sensor nodes. Suppose the source is h hops away from the sink and seeks to create a fake source on the opposite side of the sink with a similar distance to the sink. Then the source can embed that information into the data packets. As soon as the sink receives the hop count from the source, it will send a message to one of its neighbors that are in the direction of $-y$ (using the sectional antenna). This node will further pass the message to one of its neighbors whose hop count is larger than its own. If the current node that has the message does not have any neighbors with a larger hop count then we backtrace one step. We repeat this procedure until the message reaches a node whose hop count is comparable to h, and it becomes a fake source. This simple method also allows us to control the number of fake sources.

After a fake source is chosen, the rate of fake messaging can have a significant impact. Figure 2.3

presents the time series of the hunter's distance from the real source and the fake source for different fake messaging rates corresponding to $f2$ in the scenario in Figure 2.2(b). If the fake messages are injected into the network at the same rate as the real messages (as shown in Figure 2.3(i)), then the hunter oscillates between the real source and the fake source, and cannot make progress towards either of them. If the fake messages are injected at a slower rate, as shown in (ii), then the hunter will be drawn towards the real source easily. On the other hand, if the fake messaging rate is higher than the real messaging rate, then the hunter will be kept at the fake source (Figure 2.3(iii)).

The Perceptive Adversary Model: From the discussion above, one can quickly conclude that, if we have a large energy budget, we can always let fake sources inject messages at a comparable or faster speed than the real messages to protect privacy. However, this scheme cannot work for a more sophisticated hunter. By using the fact that the hunter knows that fake sources are used (Kerckhoff's Principle), the hunter may detect that he has arrived at a fake source because he cannot detect the panda. As a result, if the fake source is too close to the sink, or injects fake messages too fast, then it will be identified as a fake source quickly. Hence, it may appear appealing for the fake source to inject messages at the same rate as the real source. For the scenario in Figure Figure 2.2(b), we present the results in Figure 2.3(i), where it is seen that the hunter cannot reach either source, but just oscillate between the two. In the figure, the arrows depict the heard messages that can pull the adversary towards both the real source and the fake source. The hope is that the hunter is trapped by the two conflicting pulls into a "zigzag" movement and will not reach the real source. However, the adversary can detect the zigzag movement rather easily, with the help of its cache that stores the history of locations it has recently visited. At this point, the hunter can conclude that he might be receiving fake messages. As a response, the hunter can choose a random direction and only follow messages from that direction. In our example, let us assume that the adversary chooses to follow the messages from its right, and it can reach the fake source. As soon as it reaches the fake source, it



(a) Phantom flooding protocol

(b) Example scenario

Figure 2.4: Illustration of Phantom Flooding.

stops because the subsequent messages it receives are from the location it is at, and it can conclude it is sitting at a message source. On the other hand, the hunter is assumed to be able to detect the panda if it is at the real source. As a result, it can conclude that it has reached a fake source. Thus, it *learns* that it should only follow messages coming from its left, and can attempt to trace back to the real source. The lessons learned from the study of fake sources is that, though at an enormous energy cost, fake messaging is nonetheless not effective in protecting the privacy of source locations.

2.3.3 Phantom Routing Techniques

In the previous sections, we examined the privacy protection capabilities of baseline routing techniques and fake messaging techniques. Both approaches are not very effective in protecting privacy. In both approaches, the sources (either the real one or the fake ones) provide a fixed route for every message so that the adversary can easily back trace the route. Based on this observation, we introduce a new family of flooding and single-path routing protocols for sensor networks, called *phantom routing techniques*. The goal behind phantom techniques is to entice the hunter away from the source towards a phantom source.

In phantom routing, the delivery of every message experiences two phases: (1) the random walk phase, which may be a pure random walk or a directed walk, meant to direct the message to a phantom source, and (2) a subsequent flooding/single-path routing stage meant to deliver the message to the sink. When the source sends out a message, the message is unicasted in a random fashion for a total of h_{walk} hops. After the h_{walk} hops, in phantom flooding the message is flooded using baseline (probabilistic) flooding. In phantom single-path routing, after the h_{walk} hops the message transmission switches to single-path routing. A depiction of the phantom flooding protocol is illustrated in Figure 2.4(a).

We now discuss the random walk phase in more detail. The ability of a phantom technique to enhance privacy is based upon the ability of the random walk to place the phantom source (after h_{walk} hops) at a location far from the real source. The purpose of the random walk is to send a message to a random location away from the real source. However, if the network is more or less uniformly deployed, and we let those nodes randomly choose one of their neighbors with equal probability, then there is a large chance that the message path will loop around the source spot, and branch to a random location not far from the source.

To further quantify this notion, suppose the network of sensors $\mathcal{N}$ is arrayed on a two-dimensional integer grid with the source and asset A located at $(0, 0)$. Suppose the random walk chooses randomly from moving north, south, east, or west, i.e. from $\{(1, 0), (-1, 0), (0, 1), (0, -1)\}$, with equal probability and that the random walk may visit a node more than once. We now estimate the probability that, after h_{walk} hops, the phantom source is within a distance $d < h_{walk}$ of the true source. The movement consists of h_{walk} steps, where each step is an independent random variable X_j with vector values $\{(1, 0), (-1, 0), (0, 1), (0, -1)\}$. The location of the random walk, after h_{walk} steps, is given by

$$D_{h_{walk}} = X_1 + X_2 + \cdots + X_{h_{walk}}.$$

Then, by the central limit theorem, $D_{h_{walk}}/\sqrt{h_{walk}}$ converges in distribution to a bivariate Gaussian with mean $\mathbf{0} = (0,0)$, and covariance matrix $(1/2)\mathbf{I}$ [24]. Consequently, $D_{h_{walk}} \sim \mathcal{N}(\mathbf{0}, \frac{h_{walk}}{2}\mathbf{I})$. Let $B = B(\mathbf{0}, d)$ be a ball of radius d centered at $(0,0)$. The asymptotic probability of the phantom source's location $D_{h_{walk}}$ being within a distance d of the real source, after h random walk steps, is given by

$$\begin{aligned} P\left(D \in B\right) &= \frac{1}{h\pi} \int_B e^{-\frac{(x^2+y^2)}{h_{walk}}} dx\, dy \\ &= \frac{1}{h\pi} \int_0^d \int_0^{2\pi} e^{-r^2/h_{walk}} r\, d\theta\, dr \\ &= 1 - e^{-d^2/h_{walk}}. \end{aligned} \tag{2.1}$$

From this formula, we may examine the likelihood of the phantom's source being within 20% of h_{walk} from the true source after h_{walk} steps, i.e. $d = h_{walk}/5$. The probability is $p = 1 - e^{-h_{walk}/25}$. As we increase h_{walk}, the probability tends to 1, indicating that relative to the amount of energy spent moving a message around, we remain clustered around the true source's location. That is, purely random walk is inefficient at making the phantom source far from the real source, and therefore for reasonable h_{walk} values the location-privacy is not significantly enhanced. These results have been corroborated by simulations involving more general network arrangements, but are not presented due to space considerations.

In order to avoid random walks cancelling each other, we need to introduce bias into the walking process, and therefore we propose the use of a *directed walk* to provide location-privacy. There are two simple approaches to achieving directed walk (without equipping sensor nodes with any extra hardware) that we propose:

- *A sector-based directed random walk.* This approach requires each sensor node to be able to partition the the 2-dimensional plane into two half planes. This can be achieved without using a sectional antenna. Instead, we assume that the network field has some landmark nodes. For example, after the network is deployed, we can mark the west-most node. Then we let that node initiate a flood throughout the network. For a random node i in the network, if it forwards a packet to its neighbor j before it receives the same packet from j, then it can conclude that j is to the east; otherwise, j is to the west. Using this simple method, every node can partition its neighbors into two sets, S_0 and S_1. Before the source starts the directed random walk, it flips a coin and determines whether it is going to use S_0 or S_1. After that, within the first h_{walk} hops, every node that receives the packet randomly chooses a neighbor node from the chosen set for that packet.

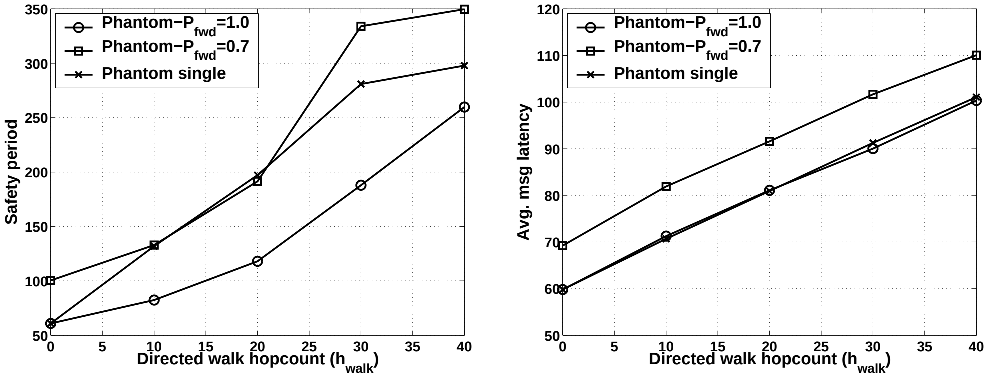
- *A hop-based directed random walk.* This approach requires each node to know the hop count between itself and the sink. This can be achieved by the sink initiating a flood throughout the network. After a node first receives the packet, it increments the hop count, and passes the packet on to its neighbors. After the flood phase, neighbors update each other with their own hop counts. As a result, node i can partition its neighbors into two sets, S_0 and S_1, where S_0 includes all the neighbors whose hop counts are smaller than or equal to i's hop count and S_1 includes all the neighbors with a larger hop count. Just as in the sector-based directed random walk, once the two sets are formed, each new message can choose a random set, and every node in the walk can choose a random neighbor from its corresponding set.

We now discuss the ability of phantom techniques to increase the safety period, and hence the location-privacy of sensor communications. Phantom flooding can significantly improve the safety period because every message may take a different (shortest) path to reach any node within the network. As a result, after the adversary hears message i, it may take a long time before it receives $i + 1$. When it finally receives message $i + 1$, the immediate sender of that message may lead the adversary farther away from the source. In the illustration shown in Figure 2.4(b), the adversary is already pretty close to the source before it receives the next new message. This new message goes through the random walk phase and reaches node A, and then goes through the flooding phase. The adversary receives this message from node B, and according to its strategy, it will be duped to move to node B, which is actually farther away from the source compared to the current location of the source.

Both phantom flooding and phantom single-path routing exhibit increased privacy protection because of the path diversity between different messages. We conducted a simulation to examine the privacy enhancement for both types of phantom routing. In this simulation, the source-sink separation was fixed at 60 hops, and we used a sector-based directed walk with different walk lengths h_{walk}. The results are presented in Figure 2.5. A value of $h_{walk} = 0$ corresponds to baseline cases. Phantom techniques clearly demonstrate a much better safety period compared to their baseline counterparts. More importantly, the improvement of phantom schemes keeps increasing with a larger h_{walk}. This is due to the fact that a larger h_{walk} creates a more divergent family of locations for the phantom source, and the probability of sending messages over precisely the same path decreases dramatically.

It is interesting to note that the safety period for phantom shortest-path is larger than for phantom flooding ($p = 1.0$). This behavior is due to the fact that, when we perform routing after the random walk, there is a high likelihood that the resulting single-paths from subsequent phantom sources will not significantly intersect and hence the hunter may miss messages. On the other hand, the resulting floods from subsequent phantom sources will still result in packets arriving at the hunter, allowing him to make progress.

The energy consumed by the phantom techniques is governed by two factors: (1) the walk distance h_{walk}, and (2) the type of flooding/single-path routing stage used. The random walk stage automatically introduces h_{walk} transmissions that were not present in the baseline cases. Typically,



(a) Safety Period

(b) Average message latency

Figure 2.5: Performance of different phantom routing techniques (source-sink separation is 60 hops).

however, the predominant energy usage for flooding-based techniques comes from the flooding phase, and usually $h_{walk} \ll n$. As a result, the increased energy consumption is negligible (in fact, it does not even change the energy consumption of baseline flooding). Further, for single-path routing techniques, it introduces at most $2h_{walk}$ extra transmissions to the shortest path between the source and the sink, and the total energy consumption of this approach is still minimal.

Phantom techniques also introduce additional latency because every message is directed to a random location first. We conducted simulations to examine the increase in latency for phantom flooding and phantom single-path routing, as presented in Figure 2.5(b). Examining this plot we see that the additional latency increases roughly linearly with h_{walk} for each phantom technique. Combining the latency results and the safety period results, it is interesting to note that for a minor increase in latency, the safety period increases dramatically. For example, for $h_{walk} = 20$, the latency increased roughly 30% while the privacy almost quadrupled!

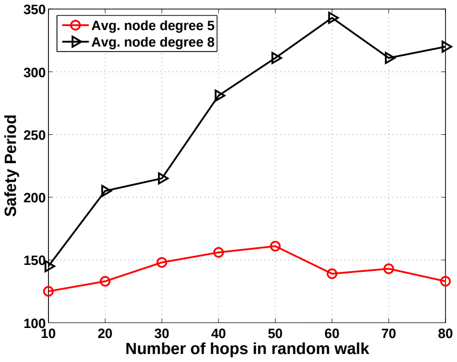


Figure 2.6: Phantom routing with different node densities

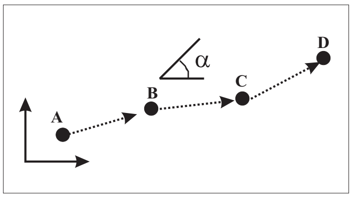


Figure 2.7: A simple movement pattern.

2.4 Impact of node density and source mobility on privacy

In this section we look at how the density of the sensor network and mobility of the source will impact the privacy protection capability of routing techniques. We focus on phantom routing performance, since it has shown the greatest promise in terms of privacy enhancement.

2.4.1 Impact of node density

The density of a sensor network is a major factor in deciding routing strategies. We characterize network density in terms of *average node degree*, which is the average number of neighbors a node has in the network. Higher node degree (dense network) means there are more alternate paths from a given node to the sink and we believe that this can be used to further enhance the source location privacy. We studied the performance of phantom shortest-path algorithm in networks with average node degree 5 and 8 respectively.

As can be seen in Figure 2.6, because of the larger number of source-sink paths available, the denser network has higher safety period. We can see that the safety period increases with increase in the h_{walk} value on the x-axis. In phantom routing, some directed walks may be in the direction of the sink and hence long h_{walk} values may end up helping the adversary by virtue of the adversary hearing the message during its directed walk phase. This is illustrated in Figure 2.6 wherein, beyond a certain point, any increase in h_{walk} does not provide any additional safety but in fact may deteriorate the safety period. This cutoff value again depends on the node density and is about 50 for the sparse network and 60(higher) for the denser network. The phantom routing technique leverages the density of the network to provide improved privacy protection.

2.4.2 Privacy Protection for a Mobile Source

In this section we study routing and the location privacy of a mobile asset A. Particularly, in the context of the Panda-Hunter Game, the panda is now mobile. The observations regarding privacy for stationary assets do not directly apply to a mobile asset scenario. Instead, a set of new questions arise. For example, since a mobile panda corresponds to a mobile source, there is a dynamically changing shortest routing path, and therefore it is natural to ask whether the moving panda alone is sufficient to protect its location privacy? Is a faster panda more safe or vice versa? How do flooding-based techniques fare for a mobile panda compared to a static one? How about single-path routing techniques?

The panda's mobility is defined by its movement pattern and its velocity. The purpose of our work is not to define a sophisticated movement pattern, nor to study a comprehensive set of movement patterns. Rather, we employ a rather simple movement model, illustrated in Figure 2.7, to study privacy. In this model, the panda knows the coordinates and knows which direction it is moving along. The parameter α governs the direction of movement. Specifically, if u is its current location, and v is its next location, then the angle of $\overrightarrow{uv}$ should be within the range $[0, \alpha]$. For instance, in Figure 2.7, the Panda traverses A, B, C, and D, and the direction of any link is within $[0, \alpha]$. Since our simulator has a finite network field, after the panda reaches the boundary of the network, it cannot find any sensor node in the specified direction, retreats a few steps, and resumes

Routing techniques	$\delta/T = 2$		$\delta/T = 6$		$\delta/T = 18$	
	L	Φ	L	Φ	L	Φ
flooding	1.0	54	1.0	50	1.0	47
phantom-flood	1.0	92	1.0	75	1.0	78
single-path	0.43	51	0.80	50	1.0	51
phantom-single	0.40	134	0.67	169	1.0	107

In this experiment the hop count between the source and the sink is 48.
The source emits a new message every 15 clock ticks.

Table 2.1: The impact of asset velocity on different routing techniques.

its normal pattern. In addition to its direction, it has the other parameters which describe its velocity: δ is the stay time at each location, and d denotes the distance for each of its movements. In the simulation, the sensor node that is closest to the Panda will become the new source, and will send $\lfloor \frac{\delta}{T} \rfloor$ (where T is the reporting interval) new messages before the Panda moves on.

The Impact of Velocity: We first conducted simulations to evaluate the effect of the panda's velocity on source-location privacy. In this experiment, the source-sink hop count was 48, and the source sends out a message every 15 clock ticks. The results are presented in Table 2.1. Here, the first observation is that, for all routing techniques, a fast moving panda (lower δ values) is safer than a slow panda. The second observation is that, among different techniques, the velocity of the panda has a more noticeable impact on single-path routing techniques than it does on flooding-based routing techniques. For single-path routing, the capture likelihood L is closely related to the velocity of the panda. In particular, a faster moving panda makes it unlikely that the adversary can track the panda. On the other hand, flooding for the same mobility allows the panda to be caught, though with an increased safety period Φ. This observation can be explained as follows. In single-path routing, subsequent shortest paths might not have significant overlap due to the panda's movement, and hence the hunter may not even see a subsequent message. On the other hand, flooding guarantees that the hunter will see the message, though not from the shortest source-sink path, and he may still follow the panda's movement. That is, a reasonably fast moving panda alone is sufficient to protect its location when using single-path routing. The third observation is that panda mobility can

```
Algorithm: Adversary Strategy II: Cautious Adversary $\mathcal{H}$

prev_location = sink;
next_location = sink;
while (next_location != source) do
    reason = TimedListen(next_location, interval);
    if (reason == MSG_ARRIVAL) then
        msg = ReceiveMessage();
        if (IsNewMessage(msg)) then
            next_location = CalculateImmediateSender(msg);
            MoveTo(next_location);
        end
    else
        next_location = prev_location;
        prev_location = LookUpPrevLocation(prev_location);
        MoveTo(next_location);
    end
end
```

Algorithm 2: *The adversary waits at a location for a period of time and returns to its previous location if no message arrives within that period of time.*

improve the privacy protection of phantom techniques more than it does to other schemes. These observations are due to the fact that the source mobility serves to further decorrelate the source's location from the phantom source's location, resulting in enhanced location privacy.

2.5 Improved Adversary Models

In this section we model three improved adversary models and study the privacy performance of phantom routing against these adversaries.

2.5.1 The Cautious Adversary Model:

We now introduce a new model for the hunter $\mathcal{H}$, which we call the *cautious adversary* model. Since phantom techniques might leave the hunter stranded far from the true source location, the cautious adversary seeks to cope by limiting his listening time at a location. If he has not received any new message within a specified interval, he concludes that he might have been misled to the current location, and he goes back one step and resumes listening from there. We illustrate the

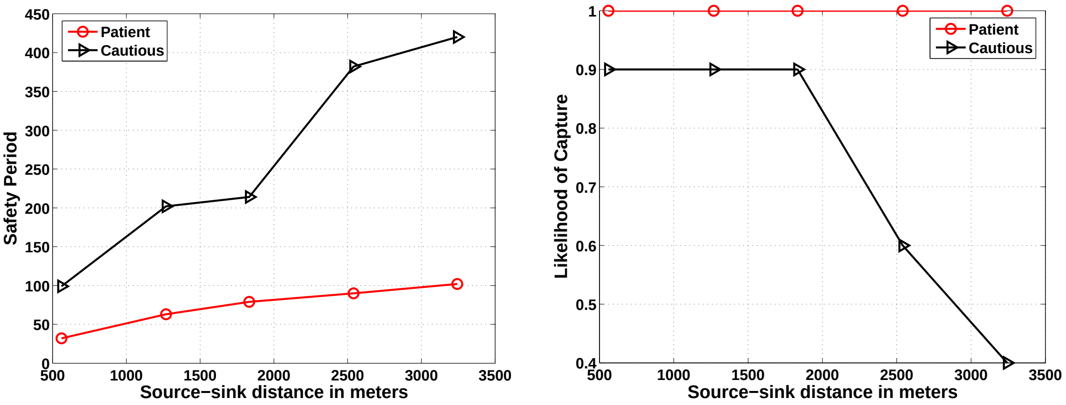


Figure 2.8: Comparing the privacy performance of phantom single-path routing for two adversarial models: the Patient and the Cautious model. ($h_{walk} = 10$ hops in these simulations)

cautious adversary model in Algorithm 2. We conducted an experiment with different source-sink separations using phantom single-path routing with $h_{walk} = 10$ hops. In our study, the cautious adversary waited at a location for a period of time corresponding to 4 source messages before deciding to retreat one step. The results are presented in Figure 2.8. The cautious adversary model does not provide any benefit over the patient adversary model. The safety period in case of the cautious adversary is consistently higher than the patient adversary, so from the adversary's point of view this is poorer performance. Further we can see that the likelihood of capture in case of the cautious adversary is also lower than the patient one and it deteriorates rapidly as the distance between the source and sink increases. The reason for this poor adversary performance is that he does not make significant forward progress with his strategy of backtracking and trying alternative paths. Consequently, it is better for the adversary to stay where he is and be patient for message to arrive.

Phantom Single-Path Routing

δ/T	$r_H/r = 1$	$r_H/r = 2$	$r_H/r = 3$
1	0.23	0.43	0.60
2	0.40	0.77	0.93
6	0.67	0.90	0.97
8	0.80	0.97	0.97

Single-path Routing

δ/T	$r_H/r = 1$	$r_H/r = 2$	$r_H/r = 3$
1	0.23	0.50	0.73
2	0.43	0.77	0.90
6	0.80	0.97	0.97
8	0.87	0.97	1.0

Table 2.2: The impact of the adversary's hearing range on capture likelihood.

2.5.2 The Adversary with improved hearing range:

So far, we have assumed that the hunter's hearing range (r_H) is the same as any normal sensor node (r). Next, let us look at the impact of different hearing ranges on the privacy level of a network. For this purpose, we conducted a set of simulation studies for phantom single-path routing with a source-sink separation of 48 hops. The resulting capture likelihoods for different δ/T and r_H/r combinations are presented in Table 2.2. In general, we find that a larger hearing range helps the hunter since this translates into the hunter hearing messages sooner and allows him to make larger moves, effectively allowing him to move faster. We also see that ability for the hunter to capture pandas improves with larger hearing ranges, and that the relative improvement is more pronounced for faster pandas. It should be realized, however, that this corresponds to introducing a powerful adversary. We also measured the impact of hearing range for single-path routing, and observed that phantom single-path routing has improved privacy for larger hearing radii compared to baseline single-path routing.

2.5.3 The Distributed Adversary:

So far we have studied scenarios with a single adversary starting at the sink and pursuing the target being tracked based on the transmissions he hears. Such an adversary is constrained by the number of transmissions that he can track at a time, limited by his hearing range. What if the adversary

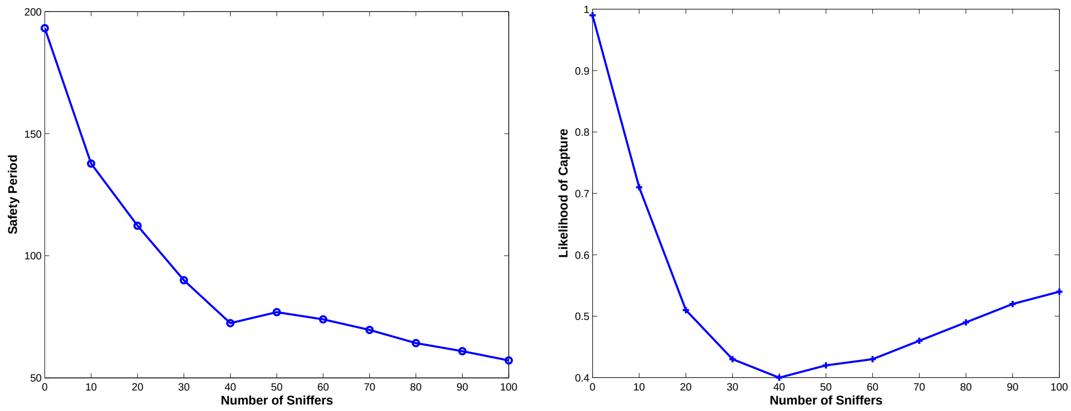


Figure 2.9: Phantom routing performance against a distributed adversary

was more powerful with the ability to monitor the action in different parts of the network simultaneously? Consider a mobile adversary who has deployed RF sniffers at various points across the sensor network and can obtain real-time readings from them. These sensors will report any new transmissions in their vicinity to the adversary. The adversary will check to see if he had heard the transmission before and if its a new transmission, he will move to the location of the sensor that reported it. We assume a powerful adversary who can instantaneously move to any sniffer location. We assume that the sniffers are distributed uniformly through the network. While it may seem like sniffers will aid the adversary in getting to the target quickly (low safety period), we found that it is not always so. In fact our simulations indicate that the patient adversary without the aid of any sniffers has better adversarial performance when phantom-shortest-path routing was used.

Figure 2.9 shows the safety period and likelihood of capture values for phantom-shortest-path routing in the presence of a distributed adversary. The setup consisted of a 10000 node network with the shortest path separation of ≈65 hops between source and sink and phantom-shortest-path routing with a 20 hop random walk being used by for routing messages. The x-axis shows the number of sniffers being used in each instance. Each point on the plot is the average of 1000 runs. The case with 0 number of sniffers is the original patient adversary model. As can be seen from the figure, for all scenarios with non-zero number of sniffers, the likelihood of capture is less than 1.0. In these cases the safety period is averaged over only in cases where the target was successfully captured by the adversary. Therefore even though the safety period for cases where sniffers are used seems to be better than the case when they are not used, one has to temper this result with the knowledge that the likelihood of a successful target capture drops rapidly as we increase the number of sniffers.

We also observe that beyond a certain threshold number of sniffers (the threshold dependent on network parameters such as topology, node density etc.) the likelihood of capture starts improving (from the adversaries point of view) albeit very very gradually. This is not surprising since beyond a certain point there are too many sniffers around to detect transmissions in every part of the network. The rate of increase of this likelihood in very gradual, meaning it would take hundreds or thousands of sniffers for the adversary to be able to achieve a likelihood of capture of close to 1.0. This is not a feasible option for the adversary because these devices are not going to be cheap. They have to have advanced communication capabilities to be able to communicate over much longer distances than the regular sensors. This is because they have to provide their own communication network consisting of other sniffers or possibly use satellite communication to communicate with the adversary in real-time. Hence deploying them in large numbers will be a costly proposition.

2.6 Related Literature

Contextual privacy issues have been examined in the context of general networks, particularly through the methods of anonymous communications. Chaum proposed a model to provide anonymity against an adversary doing traffic analysis [9]. His solution employs a series of intermediate systems called mixes. Each mix accepts fixed length messages from multiple sources and performs one or more transformations on them, before forwarding them in a random order. In the IP routing space, onion routing [6] uses this model to provide anonymous connections. Similarly, the Mixmaster remailer [10] is an email implementation of Chaum mixes. Chaum mixes provide destination privacy when an attacker knows the source. An alternative strategy to anonymity was proposed by Reiter in [8], where users are gathered into geographically diverse groups, called Crowds, to make it difficult for identifying which user makes a Web request.

In [11], a distributed anonymity algorithm was introduced that removes fine levels of detail that could compromise the privacy associated with user locations in location-oriented services. For example, a location-based service might choose to reveal that a group of users is at a specific location, or an individual is located in a vague location, but would not reveal that a specific individual is located at a specific location. Duri examined the protection of telematics data by applying privacy and security techniques [25].

Preserving privacy is an important and challenging task in data mining and databases [26–28]. A common technique is to perturb the data and to reconstruct distributions at an aggregate level. A distribution reconstruction algorithm utilizing the Expectation Maximization (EM) algorithm is discussed in [29], and the authors showed that this algorithm converges to the maximum likelihood estimate of the original distribution based on the perturbed data.

Many of these methods are not appropriate for sensor networks, particularly sensor networks that are deployed for monitoring valuable assets. In particular, location-privacy techniques built using network security mechanisms, such as the anonymity provided by mixes, incur additional communication, memory, and computational overhead that are prohibitive for use in resource-constrained environments. Consequently, full-fledged privacy solutions are not appropriate, and light-weight, resource-efficient alternatives should be explored.

Chapter 3

Temporal Privacy in Wireless Sensor Networks

It gets late early out there.

Sensor networks are being deployed to monitor a vast array of phenomena. The information surrounding these measurements can have varying levels of importance, and for this reason conventional security services, such as encryption and authentication, have been migrated to the sensor domain [3, 5, 30–32]. However, in spite of the protection that such operations might provide, there are many aspects associated with the creation and delivery of sensor messages that remain unprotected by conventional security mechanisms, and such contextual information should be protected using complimentary techniques.

Since wireless sensor networks employ a broadcast medium, an adversary may monitor sensor communications to piece together knowledge of the context surrounding sensor messages. In particular, by applying wireless localization algorithms and some level of diligence, an adversary will be able to infer the sensor network deployment, i.e. an association of sensor IDs with their physical locations. This information, combined with knowledge of the routing algorithms employed and the location of the base-station (data sink), can allow the adversary to track the spatio-temporal evolution of a sensed-event from the remote location of the network sink by merely monitoring the arrival of incoming packets [33]. This spatio-temporal information is available regardless of whether the adversary can decipher encrypted packet payloads, and represents a breach of the spatio-temporal privacy associated with the sensor network's operation. This breach of privacy can be put to very malicious use. For example, in an asset tracking sensor network, an adversary can use the spatio-temporal characteristics of the network traffic to determine the speed and direction of motion of an asset and track it down.

In order to protect against such a privacy breach, there are two types of information that can be protected: the spatial information surrounding the flow of sensor messages, and the temporal context surrounding the creation of sensor readings. Protecting the spatial context of sensor routing involves obscuring the location of the source sensor [34, 35], as well as the location of the network sink [36, 37]. However, should an adversary compromise the defense mechanisms meant to protect a sensor network's spatial context and learn the location of the originating sensor and the network sink, then the spatio-temporal context of a sensor's message flow may still be protected by employing mechanisms that protect the *temporal context* of the sensor's message.

In this chapter we focus on the problem of protecting the temporal context associated with a sensor's measurement of underlying physical phenomena. Specifically, for the typical delay-tolerant application, we propose the use of additional store-and-forward buffering at intermediate nodes along the routing path between a source sensor and the sink in order to obfuscate the time of creation associated with the flow of sensor messages.

We begin the chapter in Section 3.1 by describing our sensor network model, overview the problem of temporal privacy and how additional buffering can enhance privacy. We then examine the two conflicting aspects of buffering: in Section 3.2, we formulate temporal privacy from an information-theoretic perspective, and in Section 3.3, we examine the stress that additional delay places on intermediate buffers. Then, in Section 3.4, we present an adaptive buffering strategy that effectively manages these tradeoffs through the preemptive release of packets as buffers attain their capacity. We evaluate our temporal privacy solutions in Section 3.5 through simulations involving a large-scale network, where the adversary's mean square error is used to quantify the temporal privacy.

3.1 Overview of Temporal Privacy in Sensor Networks

We start our overview by describing a couple scenarios that illustrate the issues associated with temporal privacy. To begin, consider a sensor network that has been deployed to monitor an animal habitat [34,38]. In this scenario, animals ("assets") move through the environment, their presence is sensed by the sensor network and reported to the network sink. The fact that the network produces data and sends it to the sink provides an indication that the animal was present at the source at a specific time. If the adversary is able to associate the origin time of the packet with a sensor's location, then the adversary will be able to track the animal's behavior– a dangerous prospect if the animal is endangered and the adversary is a hunter! This same scenario can be easily translated to a tactical environment, where the sensor network monitors events in support of military networked operations. In asset tracking, if we add temporal ambiguity to the time that the packets are created then, as the asset moves, this would introduce spatial ambiguity and make it harder for the adversary to track the asset.

The situations where temporal privacy is important are not always associated with protecting spatio-temporal context, but instead there are scenarios where we are solely interested in masking the time at which an event occurred. For example, sensor networks may be deployed to monitor inventory in a warehouse. In this scenario, a sensor would create audit logs associated with the removal/relocation of items (bearing RFID tags) within the warehouse and route these audit messages to the network sink. Here, an adversary located near the sink (perhaps outside the warehouse) could observe packets arriving and use this information to infer the stock levels or the volume of transactions going through a warehouse at a specific time. Such information could be of great benefit to a rival corporation that is interested in knowing its competitor's sales and inventory profile.

Here, if we add temporal ambiguity to the delivery of the audit messages, then the warehouse would still be able to verify its inventory against purchase orders, but the competitor would have outdated information about the inventory activity.

For both scenarios, temporal privacy amounts to preventing an adversary from inferring the time of creation associated with one or more sensor packets arriving at the network sink. In order to protect the temporal context of the packet's creation, it is possible to introduce additional, random delay to the delivery of packets in order to mask a sensor reading's time of creation. However, although delaying packets might increase temporal privacy, this strategy also necessitates the use of buffering within the network and places new stress on the internal store-and-forward network buffers.

We may define a generic model for both the sensor network and the adversary that captures the most relevant features of the temporal privacy problem in this thesis. The abstract sensor network model that we will use involves:

- **Delay-Tolerant Application:** A sensor application that is delay-tolerant in the sense that observations can be delayed by reasonable amounts of time before arriving at the monitoring application, thereby allowing us to introduce additional delay in packet delivery.

- **Payload Encrypted:** The payload contains application-level information, such as the sensor reading, application sequence number and the time-stamp associated with the sensor reading. In order to guarantee the confidentiality of this data, conventional encryption is employed.

- **Headers are Cleartext:** The headers associated with essential network functionality are not encrypted. For example, the routing header associated with [39], and used in the TinyOS 1.1.7 release (described in `MultiHop.h`) includes the ID of the previous hop, the ID of the origin (used in the routing layer to differentiate between whether the packet is being generated or forwarded), the routing-layer sequence number (used to avoid loops, not flow-specific and hence cannot help the adversary in estimating time of creation), and the hop count.

On the otherhand, the assumptions that we have for the adversary are

- **Protocol-Aware:** By Kerckhoff's Principle [13], we assume the adversary has knowledge of the networking and privacy protocols being employed by the sensor network. In particular, the adversary knows the delay distributions being used by each node in the network.
- **Able to Eavesdrop:** We assume that the adversary is able to eavesdrop on communications in order to read packet headers, or control traffic. We emphasize that the adversary is not able to decipher packet contents by decrypting the payloads, and hence the adversary must infer packet creation times solely from network knowledge and the time it witnesses a packet.
- **Deployment-Aware:** We assume that the adversary at the sink and is aware of the identity of all sensor nodes. Since the adversary can monitor communications, we assume that the adversary knows the source identity associated with each transmission. Further, since the adversary is aware of the routing protocols employed and can eavesdrop, the adversary is able to build its own source-sink routing tables.
- **Non-intrusive:** The adversary does not interfere with the proper functioning of the network, otherwise intrusion detection measures might flag the adversary's presence. In particular, the adversary does not inject or modify packets, alter the routing path, or destroy sensor devices.

Taken together, we note that we have separated out issues associated with obscuring the location of the source's origin, and solely focus on temporal privacy. We note, however, that in practice the combination of temporal privacy methods with location-privacy methods will yield a more complete solution to protecting contextual privacy in sensor networks.

3.2 Temporal Privacy Formulation

We start by first examining the theoretical underpinnings of temporal privacy. Our discussion will start by first setting up the formulation using a simple network of two nodes transmitting a single packet, and then we extend the formulation to more general network scenarios.

3.2.1 Temporal Privacy: Two-Party Single-Packet Network

We begin by considering a simple network consisting of a source S, a receiver node R, and an adversarial node E that monitors traffic arriving at R. The goal of preserving temporal privacy is to make it difficult for the adversary to infer the time when a specific packet was created. Suppose that the source sensor S observes a phenomena and creates a packet at some time X. In order to obfuscate the time at which this packet was created, S can choose to locally buffer the packet for a random amount of time Y before transmitting the packet. Disregarding the negligible time it takes for the packet to traverse the wireless medium, both R and E will witness that the packet arrives at a time $Z = X + Y$. The legitimate receiver can decrypt the payload, which contains a timestamp field describing the correct time of creation. The adversary's objective is to infer the time of creation X, and since it cannot decipher the payload, it must make an inference based solely upon the observation of Z and (by Kerckhoff's Principle) knowledge of the buffering strategy employed at S.

The ability of E to infer X from Z is controlled by two underlying distributions: first, is the a priori distribution $f_X(x)$, which describes the knowledge the adversary had for the likelihood of the message creation prior to observing Z; and second, the delay distribution $f_Y(y)$, which the source employs to mask X. The amount of information that E can infer about X from observing Z is measured by the mutual information:

$$I(X;Z) \;=\; h(X) - h(X|Z) \;=\; h(Z) - h(Z|X) \;=\; h(Z) - h(Y), \tag{3.1}$$

where $h(X)$ is the differential entropy of X. For certain choices of f_X and f_Y, we may directly calculate $I(X;Z)$. For example, if $X \sim Exp(\lambda)$ (i.e. exponential with mean $1/\lambda$), and $Y \sim Exp(\lambda)$, then $Z \sim Erlang(2, \lambda)$, and $h(Z) = -\psi(2) + \ln\Gamma(2) - \ln(\lambda) + 2$, where $\psi(w)$ is the digamma function and $\Gamma(w)$ is the gamma function. For this case, $h(Y) = 1 - \ln\lambda$, and hence $I(X;Z) = 1 - \psi(2) \approx 1.077$. In other words, roughly 1 nat of information about X is learned by observing Z. For more general distributions, the entropy-power inequality [40] gives a lower bound

$$I(X;Z) \geq \frac{1}{2\ln 2}\left(2^{2h(X)} + 2^{2h(Y)}\right) - h(Y). \tag{3.2}$$

In general, however, the distribution for X is fixed and determined by an underlying physical phenomena being monitored by the sensor. Since the objective of the temporal privacy-enhancing buffering is to hide X, we may formulate the temporal privacy problem as

$$\min_{f_Y(y)} I(X;Z) = h(X+Y) - h(Y),$$

or in other words, choose a delay distribution f_Y so that the adversary learns as little as possible about X from Z.[1]

3.2.2 Temporal Privacy: Two-Party Multiple-Packet Network

We now extend the formulation of temporal privacy to the more general case of a source S sending a stream of packets to a receiver R in the presence of an adversary E. In this case, the sender S will create a stream of packets at times $X_1, X_2, \ldots, X_n, \ldots$, and will delay their transmissions by $Y_1, Y_2, \ldots, Y_n, \ldots$. The packets will be observed by E at times $Z_1, Z_2, \ldots, Z_n, \ldots$. In going to the

[1] The astute reader will note the similarity with the information-theoretic formulation of communication, where the objective is to maximize mutual information.

more general case of a packet stream, several new issues arise. First, as noted earlier in Section 3.1, when we delay multiple packets it will be necessary to buffer these packets. For now we will hold off on discussing queuing issues until Section 3.3. The next issue involves how the packets should be delayed. There are many possibilities here. For example, one possibility would have packets released in the same order as their creation, i.e. $Z_1 < Z_2 < \ldots < Z_n$, which would correspond to choosing Y_j to be at least the wait time needed to flush out all previous packets. Such a strategy does not reflect the fact that most sensor monitoring applications do not require that packet ordering is maintained. Therefore, a more natural delay strategy would involve choosing Y_j independent of each other and independent of the creation process $\{X_j\}$. Consequently, there will not be an ordering of $(Z_1, Z_2, \ldots, Z_n, \ldots)$.

In our sensor network model, however, we assumed that the sensing application's sequence number field was contained in the encrypted payload, and consequently the adversary does not directly observe $(Z_1, Z_2, \ldots, Z_n, \ldots)$, but instead observes the sorted process $\{\tilde{Z}_j\} = \Upsilon(\{Z_j\})$, where $\Upsilon(\{Z_j\})$ denotes the permutations needed to achieve a temporal ordering of the elements of the process $\{Z_j\}$, i.e. $\{\tilde{Z}_j\} = (\tilde{Z}_1, \tilde{Z}_2, \ldots, \tilde{Z}_n, \ldots)$ where $\tilde{Z}_1 < \tilde{Z}_2 < \cdots$. The adversary's task thus becomes inferring the process $\{X_j\}$ from the sorted process $\{\tilde{Z}_j\}$. The amount of information gleaned by the adversary after observing $\tilde{Z}^n = (\tilde{Z}_1, \cdots, \tilde{Z}_n)$ is thus $I(X^n; \tilde{Z}^n)$, and the temporal-privacy objective of the system designer is to make $I(X^n; \tilde{Z}^n)$ small.

Although it is analytically cumbersome to access $I(X^n; \tilde{Z}^n)$, we may use the data processing inequality [2] [40] on $X^n \to Z^n \to \tilde{Z}^n$ to obtain the relationship $0 \leq I(X^n, \tilde{Z}^n) \leq I(X^n, Z^n)$, which allows us to use $I(X^n, Z^n)$ in a pinching argument to control $I(X^n, \tilde{Z}^n)$. Expanding $I(X^n, Z^n)$

[2] The data processing inequality: If $X \to Y \to Z$ then $I(X, Y) \geq I(X, Z)$

as

$$\begin{aligned} I(X^n, Z^n) &= h(Z^n) - h(Y^n) \\ &\leq \sum_{j=1}^{n} (h(Z_j) - h(Y_j)) \\ &= \sum_{j=1}^{n} I(X_j, Z_j), \end{aligned} \tag{3.3}$$

we may thus bound $I(X^n, Z^n)$ using the sum of individual mutual information terms.

As before, the objective of temporal privacy enhancement is to minimize the information that the adversary gains, and hence to mask $\{X_j\}$, we should minimize $I(X^n, Z^n)$. Although there are many choices for the delay process $\{Y_j\}$, the general task of finding a non-trivial stochastic process $\{Y_j\}$ that minimizes the mutual information for a specific temporal process $\{X_j\}$ is challenging and further depends on the sensor network design constraints (e.g. buffer storage). In spite of this, however, we may seek to optimize within a specific type of process $\{Y_j\}$, and from this make some general observations.

As an example of this, let us look at an important and natural example. Suppose that the source sensor creates packets at times $\{X_j\}$ as a Poisson process of rate λ, i.e. the interarrival times A_j are exponential with mean $1/\lambda$, and that the delay process $\{Y_j\}$ corresponds to each Y_j being an exponential delay with mean $1/\mu$. One motivation for choosing an exponential distribution for the delay is the well-known fact that the exponential distribution yields maximal entropy for non-negative distributions. We note that $X_j = \sum_{k=1}^{j} A_k$ (and hence the X_j are j-stage Erlangian random variables with mean j/λ). Using the result of Theorem 3(d) from [41], we have that

$$\begin{aligned} I(X_j; Z_j) &= I(X_j; X_j + Y_j) \\ &= \ln\left(1 + \frac{j\mu}{\lambda}\right) - D\left(f_{X_j+Y_j} \| f_{\overline{X}_j+Y_j}\right) \\ &\leq \ln\left(1 + \frac{j\mu}{\lambda}\right). \end{aligned} \tag{3.4}$$

Here, the $D(f\|g)$ corresponds to the divergence between two distributions f and g, while $\overline{X}$ is the mixture of a point mass and exponential distribution with the same mean as X, as introduced in [41]. Since divergence is non-negative and we are only interested in pinching $I(X^n; \tilde{Z}^n)$, we may discard this auxiliary term. Using the above result, we have that

$$I(X^n, Z^n) \;\leq\; \sum_{j=1}^{n} \ln\left(1 + \frac{j\mu}{\lambda}\right). \tag{3.5}$$

Our objective is to make

$$0 \;\leq\; I(X^n; \tilde{Z}^n) \;\leq\; I(X^n, Z^n) \;\leq\; \sum_{j=1}^{n} \ln\left(1 + \frac{j\mu}{\lambda}\right)$$

small, and from this we can see that by tuning μ to be small relative to λ (or equivalently, the average delay time $1/\mu$ to be large relative to the average interarrival time $1/\lambda$), we can control the amount of information the adversary learns about the original packet creation times. It is clear that choosing μ too small will place a heavy load on the source's buffer. We will revisit buffer issues in Section 3.3 and Section 3.4.

3.2.3 Temporal Privacy: Multihop Networks

In the previous subsection, we considered a simple network case consisting of two nodes, where the source performs all of the buffering. More general sensor networks consist of multiple nodes that communicate via multi-hop routing to a sensor network sink. For such networks, the burden of obfuscating the times at which a source node creates packets can be shared amongst other nodes on the path between the source and the sensor network sink.

To explain, we may consider a generic sensor network consisting of an abundant supply of sensor nodes, and focus on an N-hop routing path between the source and the network sink. By doing so, we are restricting our attention to a line-topology network $S \to F_1 \to F_2 \to \cdots \to F_{N-1} \to R$, where R denotes the receiving network sink, and F_j denotes the j-th intermediate node on the forwarding path.

By introducing multiple nodes, the delay process $\{Y_j\}$ can be decomposed across multiple nodes as

$$Y_j = Y_{0j} + Y_{1j} + \cdots + Y_{N-1,j},$$

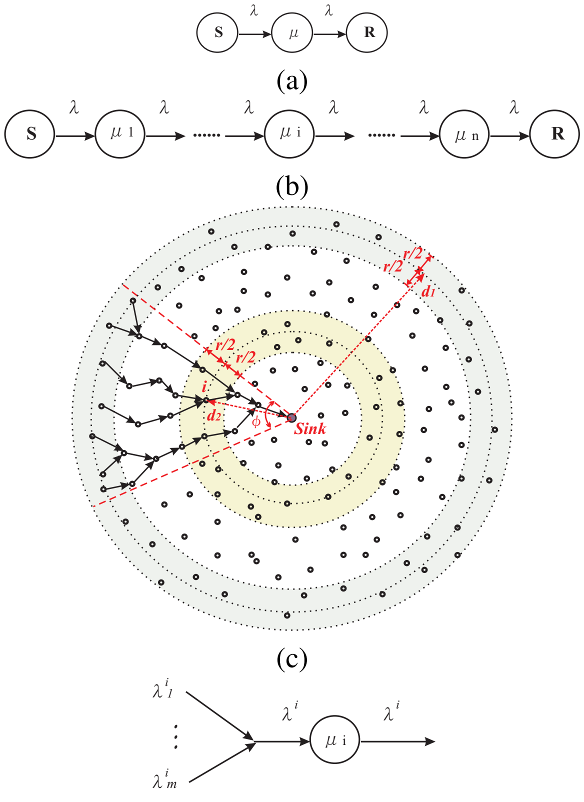
where Y_{kj} denotes the delay introduced at node k for the j-th packet (we use Y_{0j} to denote the delay used by the source node S). Thus, each node k will buffer each packet j that it receives for a random amount of time Y_{kj}.

This decomposition of the delay process $\{Y_j\}$ into sub-delay processes $\{Y_{kj}\}$ allows for great flexibility in achieving both temporal privacy goals and ensuring suitable buffer utilization in the sensor network. For example, it is well-known that traffic loads in sensor networks accumulate near network sinks, and it may be possible to decompose $\{Y_j\}$ so that more delay is introduced when a forwarding node is further from the sink.

3.3 Queuing Analysis of Privacy-Enhancing Buffering

Although delaying packets might increase temporal privacy, such a strategy places a burden on intermediate buffers. In this section we will examine the underlying issues of buffer utilization when employing delay to enhance temporal privacy.

When using buffering to enhance temporal privacy, each node on the routing path will receive packets and delay their forwarding by a random amount of time. As a result, sensor nodes must buffer packets prior to releasing them, and we may formulate the buffer occupancy using a queuing model. In order to start our discussion, let us again examine the simple two-node case where a source node S generates packets according to an underlying process and the packets are delayed according to an exponential distribution with average delay $1/\mu$, prior to being forwarded to the receiver R, as depicted in Figure 3.1 (a). If we assume that the creation process is Poisson with



(a)

(b)

(c)

(d)

Figure 3.1: (a) Queue model for buffering at the source node S, (b) chain of queues along a routing path from S to receiver sink R, (c) the effect of flow convergence in a large sensor network, and (d) queuing model for the merging of traffic flows at an intermediate sensor node.

rate λ (if the process is not Poisson, the source may introduce additional delay to shape the traffic), then the buffering process can be viewed as an $M/M/\infty$ queue where, as new packets arrive at the buffer, they are assigned to a new "variable-delay server" that processes each packet according to an exponential distribution with mean $1/\mu$. Following the standard results for $M/M/\infty$ queues, we have that the amount of packets being stored at an arbitrary time, $N(t)$, is Poisson distributed, with $p_k = P\{N(t) = k\} = \frac{\rho^k}{k!}e^{-\rho}$, where $\rho = \lambda/\mu$ is the system utilization factor. $\overline{N}$, the expected number of messages buffered at S, is ρ.

The slightly more complicated scenario involving more than one intermediate node allows for the buffering responsibility to be divided across the routing path, and is depicted by a chained path in Figure 3.1 (b). A tandem queuing network is formed, where a message departing from node i immediately enters an $M/M/\infty$ queue at node $i+1$. Thus, the interdeparture times from the former generate the interarrival times to the latter. According to Burke's Theorem [42], the steady-state output of a stable $M/M/m$ queue with input parameter λ and service-time parameter μ for each of the m servers is in fact a Poisson process at the same rate λ when $\lambda < \mu$. Hence, we may generally model each node i on the path as an $M/M/\infty$ queue with average input message rate λ, but with average service-time $1/\mu_i$ (to allow each node to follow its own delay distribution).

So far we have only considered a single routing path in a sensor network, but in practice the network will monitor multiple phenomena simultaneously, and consequently there will be multiple source-sink flows traversing the network. As a result, for the most general scenario, the topological structure of the network will have an impact on buffer occupancy. For example, nodes that are closer to network sink typically have higher traffic loads, and thus will be expected to suffer from a higher buffer occupancy than nodes further from the sink. We now explore this behavior, and the relationship between buffering for privacy-enhancement and the traffic load placed on intermediate nodes due to flow convergence in the sensor network.

Consider a sensor network deployment as depicted in Figure 3.1(c), where we have assumed (without loss of generality) that there is only one sink. Here, multiple sensors generate messages intended for the sink, and each message is routed in a hop-by-hop manner based on a routing tree (as suggested in the figure). Message streams merge progressively as they approach the sink. If we assume that the senders in the network generate Poisson flows, then by the superposition property of Poisson processes, the combined stream arriving at node i of m independent Poisson processes with rate λ_j^i is a Poisson process with rate $\lambda^i = \lambda_1^i + \lambda_2^i + \cdots + \lambda_m^i$. We depict this phenomena for node i in Figure 3.1(d), where m is the number of "routing" children for node i. Additionally, we let $1/\mu_i$ be the average buffer delay injected by node i. Then node i is an $M/M/\infty$ queue, with arrival parameter λ^i and departure parameter μ_i, yielding:

- $N_i(t)$, the number of packets in the buffer at node i, is Poisson distributed.
- $p_{ik} = P\{N_i(t) = k\} = \frac{\rho_i^k}{k!} e^{-\rho_i}$, where $\rho_i = \lambda^i/\mu_i$.
- The expected number of messages at node i is $\overline{N_i} = \rho_i$.

As expected, if we choose our delay strategy at node i such that μ_i is much smaller than λ^i (as is desirable for enhanced temporal privacy), then the expected buffer occupancy $\overline{N_i}$ will be large. Thus, temporal privacy and buffer utilization are conflicting system objectives.

We now evaluate the impact of the depth of node i in the routing tree (the number of hops from the node i to the sink). For the sake of calculations, we shall assume that the density η of the sensor deployment is sufficient that a communicating sensor node will always find a path to the network sink. Additionally, let us denote the average geographical distance between parents and children in the routing tree by r. Then, to quantify the effect of flow convergence on the local traffic rate in the sensor network, let us assume that an outer annulus O_1 of distance d_1, angular spread φ and width r creates a total traffic of rate λ_{O_1} packets/second, as depicted in Figure 3.1 (c). Hence, in a spatial ensemble sense, each node carries an average traffic rate of $\overline{\lambda_{O_1}} = \lambda_{O_1}/(\varphi r d_1 \eta)$. This traffic flows toward the sink, and if we examine an annulus at distance $d_2 < d_1$ with width r and spread φ, the area of this annulus is $\varphi r d_2$, and there will be an average of $\varphi r d_2 \eta$ sensors in O_2 carrying a total rate of λ_{O_1}. Hence, on average, each sensor in this inner annulus will carry traffic of rate $\overline{\lambda_{O_2}} = \lambda_{O_1}/(\varphi r d_2 \eta)$. Comparing the average traffic load $\overline{\lambda_{O_2}}$ that a single sensor in an inner annulus O_2 carries with the average traffic load $\overline{\lambda_{O_1}}$ of a single sensor in annulus O_1, yields

$$\frac{\overline{\lambda_{O_2}}}{\overline{\lambda_{O_1}}} = \frac{d_1}{d_2}, \tag{3.6}$$

and hence traffic load increases in inverse relationship to the distance a node is from the sink.

The last issue that we need to consider is the amount of storage available for buffering at each sensor. As sensors are resource-constrained devices, it is more accurate to replace the $M/M/\infty$ queues with $M/M/k/k$ queues, where memory limitations imply that there are at most k servers/buffer slots, and each buffer slot is able to handle 1 message. If an arriving packet finds all k buffer slots full, then either the packet is dropped or, as we shall describe later in Section 3.4, a preemption strategy can be employed. For now, we just consider packet dropping. We note that packet dropping at a single node causes the outgoing process to lose its Poisson characteristics. However, we further note that by Kleinrock's Independence approximation (the merging of several packet streams has an affect akin to restoring the independence of interarrival times) [42], we may continue to approximate the incoming process at node i as a Poisson process with aggregate rate λ^i. Hence, in the same way as we used a tree of $M/M/\infty$ queues to model the network earlier, we can instead model the network as a tree of $M/M/k/k$ queues.

The $M/M/k/k$ formulation provides us with a means to adaptively design the buffering strategy at each node. If we suppose that the aggregate traffic levels arriving at a sensor node is λ, then the packet drop rate (the probability that a new packet finds all k buffer slots full) is given by the well-known Erlang Loss formula for $M/M/k/k$ queues:

$$E(\rho, k) \quad = \quad \frac{\rho^k}{k!} p_0 \quad = \quad \frac{\frac{\rho^k}{k!}}{\sum_{i=0}^{k} \frac{\rho^i}{i!}}, \tag{3.7}$$

where $\rho = \lambda/\mu$. For an incoming traffic rate λ, we may use the Erlang Loss formula to appropriately select μ so as to have a target packet drop rate α when using buffering to enhance privacy. This observation is powerful as it allows us adjust the buffer delay parameter μ at different locations in the sensor network, while maintaining a desired buffer performance. In particular, the expression for $E(\rho, k)$ implies that, as we approach the sink and the traffic rate λ increases, we must decrease the average delay time $1/\mu$ in order to maintain $E(\rho, k)$ at a target packet drop rate α.

3.4 RCAD: Rate-Controlled Adaptive Delaying

A consequence of the results of the previous section is that nodes close to the sink will have high buffer demands and their buffers may be full when new packets arrive. In practice, we need to adjust the delay distribution as a function of the incoming traffic rate and the available buffer space.

In order to accomplish this adjustment, we propose *RCAD*, a Rate-Controlled Adaptive Delaying mechanism, to achieve privacy and desirable performance simultaneously. The main idea behind RCAD is buffer preemption– if the buffer is full, a node should select an appropriate buffered packet, called the *victim packet*, and transmit it immediately rather than drop packets. Consequently, preemption automatically adjusts the effective μ based on buffer state. In this thesis, we have proposed the following buffer preemption policies:

- *Longest Delayed First (LDF).* In this policy, the victim packet is the packet that has stayed in the buffer the longest. By doing so, we can ensure that each packet is buffered for at least a short duration. The implementation of this policy requires that each node record the arrival time of every packet.

- *Longest Remaining Delay First (LRDF).* In this policy, the victim packet is the packet that has the longest remaining delay time. Preempting such packets can lessen the buffer load more than any other policy because such packets would have resided in the buffer the longest. The implementation of this scheme is straightforward because each node already keeps track of the remaining buffer time for every packet.

- *Shortest Delay Time First (SDTF).* In this policy, the victim packet is the one with the shortest delay time. By lessening an already short delay time, we expect that the overall performance will remain roughly the same. The implementation of this policy requires each node record the delay of every packet.

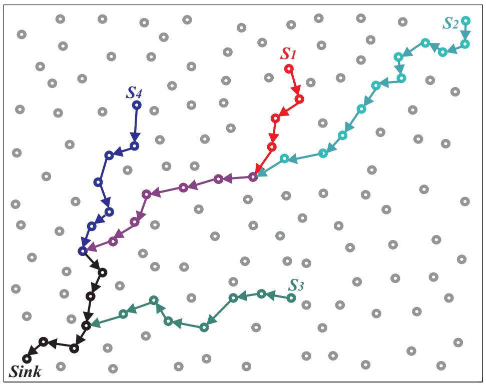


Figure 3.2: Simulation topology

- *Shortest Remaining Delay First (SRDF).* In this policy, the victim packet is the packet that has the shortest remaining delay time. In this way, the resulting delay times for that node are the closest to the original distribution. As in the case of the LRDF policy, the implementation is straightforward.

3.5 Evaluating RCAD Using Simulations

In this study, we have developed a detailed event-driven simulator to study the performance of RCAD. The simulations modeled realistic network/traffic settings, and measured important performance and privacy metrics.

3.5.1 Performance Metrics and Adversary Models

In our simulated sensor network, we have multiple source nodes that create packets, and intermediate nodes that follow RCAD schemes for buffering packets prior to forwarding them. As an important player of the game, the adversary stays at the sink, observes packet arrivals, and estimates the creation times of these packets.

In this study, we assume a powerful adversary that can acquire the following parameters for each flow: (1) the hop count of that flow, (2) the delay distributions for nodes along the flow, and (3) the traffic arrival process of the flow, e.g. the arrival rate, the arrival distribution, etc. For an observed packet arrival time z, a baseline adversary estimates the creation time of this packet as $x' = z - y$, where y is the average delay of the flow, which the adversary can calculate from its knowledge of the delay distributions. In the simulations, we use the *square error* to quantify the estimation error, i.e. $(x' - x)^2$ where x is the true creation time. Similarly, for a series of packet arrivals from the same flow $z_1, z_2, \ldots, z_m$, a baseline adversary estimates their creation times as $x'_1, x'_2, \ldots, x'_m$, and $x'_i = z_i - y$. The total estimation error for m packets is then calculated as the *mean square error*, $\sum(x'_i - x_i)^2/m$. We note that there is a direct relationship between mutual information and mean square error [43], and hence the scheme that has a higher estimation error consequently better preserves the temporal privacy of the source.

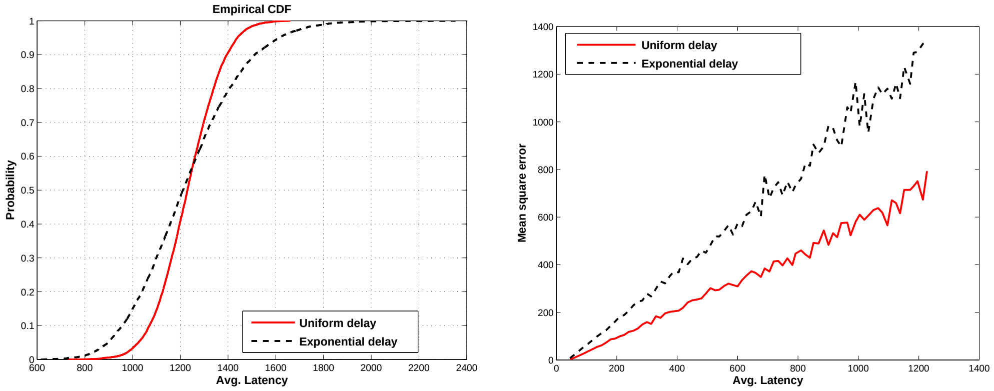
Since RCAD schemes dynamically adapt the delay processes by adopting buffer preemption strategies, it is inadequate for the adversary to estimate the actual delay times using the original delay distributions before preemption. As a result, we also enhance the baseline adversary to let the adversary adapt his estimation of the delays. We call such an adversary as an *adaptive* adversary.

In order to understand our adaptive adversary model, let us first look at a simple example. Let us assume there is only one node with one buffer slot between the source and sink. Further, assume that the packet arrival follows a Poisson process with rate λ, and the buffer generates a random delay time that follows an exponential distribution with mean $1/\mu$. If the buffer at the intermediate node is full when a new packet arrives, the currently buffered packet will be transmitted. In this example, if the traffic rate is low, say $\lambda < \mu$, then the packet delay time will be $1/\mu$. However, as the traffic increases, the average delay time will become $1/\lambda$ due to buffer preemptions. Following this example, our adaptive adversary should adopt a similar estimation strategy: at low traffic rates, he estimates the overall average delay y by h/μ, while at higher traffic rates, he estimates the overall average delay y as a function of the buffer space and the incoming rate, i.e. hk/λ, where h is the flow hop count, k is the number of buffer slots at each node, and λ is the traffic rate of that flow. Given an aggregated traffic rate λ_{tot} from n sources converging at least one-hop prior to the sink, the adversary can compute the probability of buffer overflow via the Erlang Loss formula in equation (3.7). He then can compare this against a chosen threshold and if the probability is less than the threshold, he will assume the average delay introduced by each hop is $1/\mu$. However, if the probability is higher than the threshold, the average delay at each node is calculated to be nk/λ_{tot}.

Additionally, we note that it is desirable to achieve privacy while maintaining tolerable end-to-end delivery latency for each packet. Hence, in our studies, for a network performance metric we use the average end-to-end delivery latency for packets coming from a particular flow versus the underlying traffic rate and the RCAD strategies employed.

3.5.2 Simulation Setup

The topology that we considered in our simulations is illustrated in Figure 3.2. Here, nodes S_1, S_2, S_3, and S_4 are source nodes and create packets that are destined for the sink. Thus, we had four flows, and these flows had hop counts 15, 22, 9 and 11 respectively. Each source generated a total of 1000 packets with a mean interarrival time of $1/\lambda$ time units. In our experiments we varied $1/\lambda$ from 2 (i.e. the highest traffic rate) time units to 20 (the slowest traffic rate) to generate different cases of traffic loads for the network. The main focus of our simulator is the scale of the network, so we simplified the PHY- and MAC-level protocols by adopting a constant transmission delay (i.e. 1 time unit) from any node to its neighbors. When a packet arrives at an intermediate node, the intermediate node introduces a random delay following an exponential distribution with mean $1/\mu$. Unless mentioned otherwise we took $1/\mu = 30$ time units in the simulations. The results reported are for the flow S_1 to the sink.



(a) Average Latency

(b) Average Mean Square Error

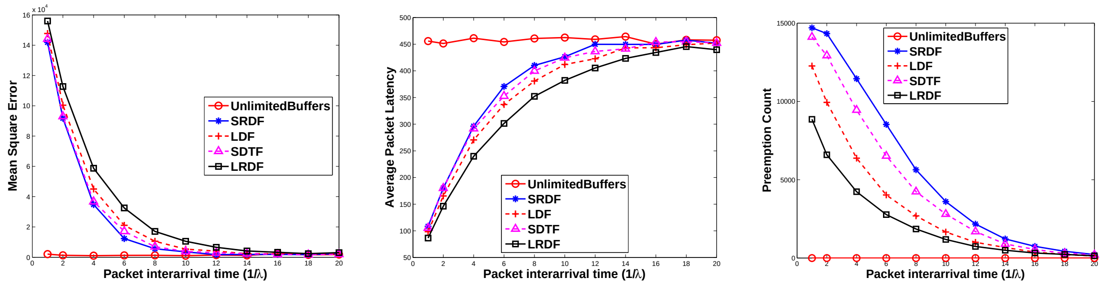
Figure 3.3: Comparing expoential delay distribution to uniformly random delay distribution

3.5.3 Performance Results

Before analyzing the performance of RCAD strategies, we illustrate how choosing exponential delay distribution achieves a better tradeoff between overall message latency and offering better uncertainty as compared to uniformly random delay distribution. Figure 3.3(a) shows the cdf of average end-to-end latency for both the distributions. As we can see they are pretty close to each other in terms of latency but the exponential delay distribution generates much larger error in an adversary's estimate of the time of origin of a message for the same average latency.

Comparison of RCAD Strategies

Figures 3.4(a) and (b) present the mean square error and the delivery latency of the four RCAD strategies where we assume each sensor node has a buffer of 10 slots (which is typical for a Mica2 mote), and a preemption-less strategy that assumes unlimited buffer space on each node. In this set of experiments, we used the baseline adversary model that estimated the delay for flow i as h_i/μ, where h_i is the hop count of flow i and $1/\mu$ is the average per hop delay (30 time units). At low traffic rates ($1/\lambda = 16, 18, 20$), these five strategies perform the same because the average



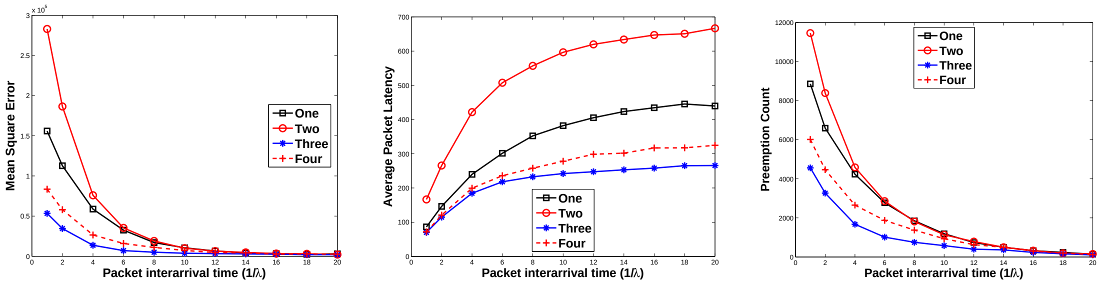
(a) Mean square error (b) Delivery latency (c) Number of preemptions

Figure 3.4: Comparison of the four RCAD strategies and the scenario with unlimited buffers. We note the x-axis is in terms of average source interarrival time $1/\lambda$.

buffer requirement per node is less than 10. As the traffic increases, the four preemption strategies lead to much higher mean square error, thus providing better temporal privacy. This is because the baseline adversary did not take into consideration the effect of buffer preemptions. Among the four RCAD strategies, we observed that LRDF policy consistently performs the best in terms of privacy, followed by LDF and SDTF, while SRDF was the worst.

In order to understand the difference between these four strategies, let us look at more detailed statistics. Figure 3.4(c) presents the number of preemptions that occurred during the experiments. We observe the opposite order here: the strategy that provides the most privacy incurred the least number of preemptions. This may appear counter-intuitive at first glance, but can be simply explained: the strategy that leads to more preemptions tends to alter the original delay distribution less, and thus confuses the adversary less. For example, LRDF selects the packet that has the longest remaining delay time as the victim packet. Preempting these packets will have two effects: (1) it will alter the original delay distribution more, and (2) it will reduce the number of preemptions.

Moving our attention to delivery latency, we observed that the preemption-less strategy with unlimited buffers incurred much longer latencies at higher traffic rates. Among the four preemption strategies, LRDF has the shortest latency because it tends to reduce the delay times in the buffer the



(a) Mean square error (b) Delivery latency (c) Number of preemptions

Figure 3.5: Comparison of the LRDF algorithm for four different source-sink distances. We note the x-axis is in terms of average source interarrival time $1/\lambda$.

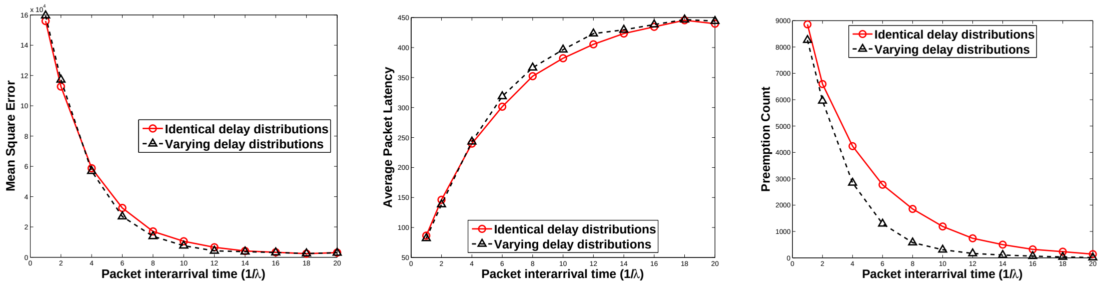
most.

Impact of distance of a source from the adversary

Figure 3.5 shows the impact distance has on the performance of RCAD algorithms. We compare the same metrics as above for four different sources as shown in Figure 3.2 and using LRDF. Note here that this comparison also allows us to answer the question of how RCAD would work if the adversary had compromised a fraction of the nodes in the network and programmed them never to delay any packets. We can think of this act as equivalent of the adversary reducing the number of hops between the source and the sink and hence get an idea of the impact of such a compromise on the temporal privacy using graphs similar to that shown in Figure 3.5.

Reducing Preemption by Adopting Varying Delay Distributions

Buffer preemption is necessary to avoid dropping packets due to buffer saturation, and we have just seen that it can help provide better temporal privacy. Buffer preemption, however, also has disadvantages, especially as it introduces additional protocol overhead at each sensor node associated with the selection of victim packets. Hence, in order to reduce protocol overhead we must reduce the frequency of preemption, but at the same time strive to maintain the same level of temporal



(a) Mean square error (b) Delivery latency (c) Number of Preemptions

Figure 3.6: Comparison of the performance of the LRDF policy when all nodes have identical delay distributions and when nodes have varying delay distributions.

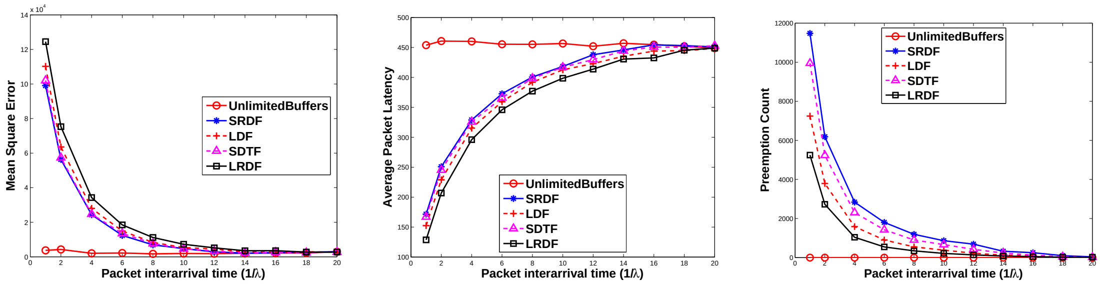
privacy.

One strategy for reducing preemption is to let each node employ a different delay distribution. Since sensor networks usually have many more sources than sinks, as illustrated in Figure 3.1 (c), nodes closer to the sink experience higher traffic volumes than nodes closer to the source. As a result, the nodes closer to the sink should delay packets much less in order to relieve the buffer requirements at these nodes. Our objective with this approach is to keep the buffer usage the same across all the nodes. As discussed in Section 3.3, the number of buffered packets at node i can be estimated as $\overline{N_i} = \rho_i = \frac{\lambda_i}{\mu_i}$. Suppose we consider a flow with h hops (i.e. h nodes before the sink), and use node 1 to denote the last node before the sink and node h to denote the source node. To keep $\overline{N_i}$ constant across all nodes while having a target overall average delay of D, we choose the average delay time $1/\mu_i$ for node i as β/h_i, where β is the coefficient and h_i is the hop count between node i and the sink. Thus, we have

$$\sum_{i=1}^{h} \frac{\beta}{h_i} = \sum_{i=0}^{h-1} \frac{\beta}{i+1} = \beta(\gamma + \psi(h+1)), \tag{3.8}$$

where γ is the Euler-Mascheroni constant and $\psi(x)$ is the digamma function. Hence, the average delay time $1/\mu_i$ for node i is calculated as

$$1/\mu_i = \frac{D}{(i+1)(\gamma + \psi(h+1))}. \tag{3.9}$$



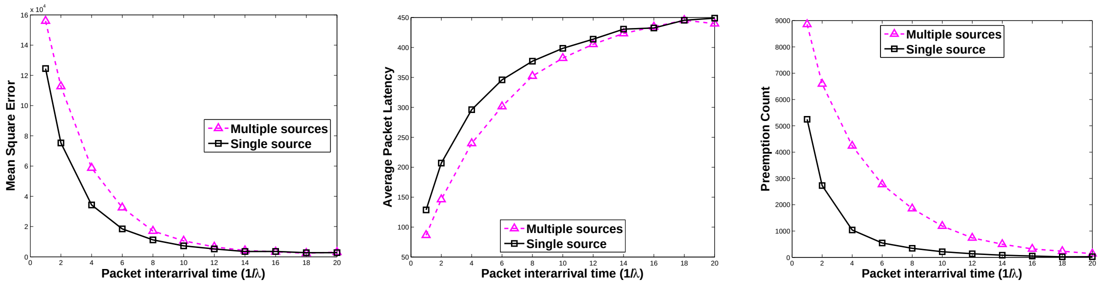
(a) Mean square error (b) Delivery latency (c) Number of Preemptions

Figure 3.7: RCAD Performance with a single source and no aggregation of traffic

We conducted a set of experiments to study the performance of RCAD strategies when using varying delay distributions chosen as above. The results with LRDF are presented in Figure 3.6. We observe that employing variable delays can significantly reduce the number of preemptions, especially for mid-range traffic rates. At the same time, having variable delays will not degrade either the mean square error or latency much. Although the preemptions were reduced by an amount up to 70%, the largest estimation error reduction we observed was 16%, while the largest latency increase was only 4%.

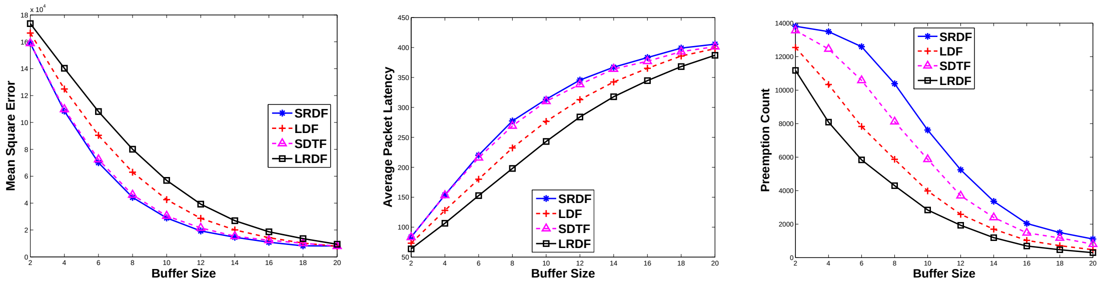
Impact of aggregation

Figure 3.7 demonstrates the RCAD performance in the presence of a single source of traffic and therefore no aggregation taking place at any node. We can see that the performance follows a trend similar to that in the presence of multiple sources with LRDF coming up the winner. Figure 3.8 shows head-to-head comparison of privacy protection offered to source in our topology in the presence or absence of other simultaneous sources of traffic. We can see that RCAD provides quite comparable performance even to a single stream.



(a) Mean square error (b) Delivery latency (c) Number of Preemptions

Figure 3.8: Comparison of the performance of the LRDF policy for single and multiple sources



(a) Mean square error (b) Delivery latency (c) Number of Preemptions

Figure 3.9: RCAD performance with varying buffer sizes

Impact of buffer size

Figure 3.9 shows the performance of RCAD algorithms with varying buffer sizes on the nodes. We can see that RCAD algorithms behave very well with low buffer sizes in-fact better than when they have large buffers at their disposal. We hinted at this in earlier discussion but show concrete proof here. This happens because the smaller buffer sizes mean the RCAD algorithms deviate from the mean delay of their exponential delay distribution and thus the adversary is thrown off-base from his estimate which uses this mean delay.

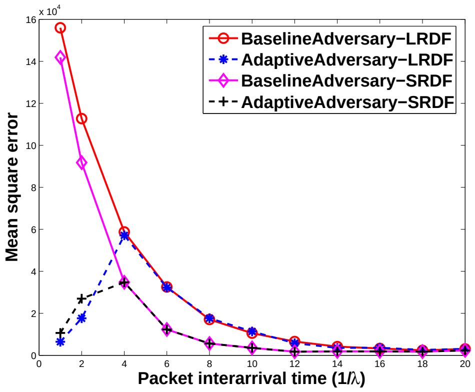


Figure 3.10: The estimation error for the two adversary models, when LRDF and SRDF RCAD are employed.

The Adaptive Adversary Model

A baseline adversary is inefficient in estimating delays for RCAD strategies with preemption. As a result, we studied the ability of an adaptive adversary to estimate the time of creation when using RCAD with identical delay distributions across the network. The resulting estimation mean square errors are presented in Figure 3.10. The adaptive adversary adopts the same estimation strategy as the baseline adversary at lower traffic rates, i.e. $1/\mu$ per hop, but it uses the incoming traffic rate to estimate the delay at higher traffic rates, i.e. $h_i k/\lambda_i$ for the average delay of flow i. The switch between estimation strategies used the Erlang Loss formula for a threshold preemption rate of 0.1. Figure 3.10 shows that the adaptive adversary can significantly reduce the estimation errors, especially at higher traffic rates (lower interarrival times) where preemption is more likely.

An interesting observation is that at high traffic rates, the adaptive adversary can more accurately estimate the delay times generated by the LRDF policy when compared to other RCAD policies. Recall that earlier, LRDF had the highest estimation error against a baseline adversary while the SRDF policy had the least error (and hence the least privacy). Now, at high traffic rates against an adaptive adversary, the trend is reversed– SRDF has the best privacy while LRDF has the worst.

3.6 Related Work

The problem of preserving privacy has been considered in the context of data mining and databases [26–28]. A common technique is to perturb the data and to reconstruct distributions at an aggregate level. A distribution reconstruction algorithm utilizing the Expectation Maximization (EM) algorithm is discussed in [29], and the authors showed that this algorithm converges to the maximum likelihood estimate of the original distribution based on the perturbed data.

Contextual privacy issues have been examined in general networks, particularly through the methods of anonymous communications. Chaum proposed a model to provide anonymity against an adversary conducting traffic analysis [9]. His solution employs a series of intermediate systems called mixes. Each mix accepts fixed length messages from multiple sources and performs one or more transformations on them, before forwarding them in a random order. Most of the early mix related research was done on *pool mixes* [44], which wait until a certain threshold number of packets arrive before taking any mixing action. Kesdogan [45] proposed a new type of mix, *SG-Mix*, which delays an individual incoming message according to an exponential distribution before forwarding them on. Later, Danezis proved in [46] using information theory that a SG-Mix is the optimal mix strategy that maximizes anonymity. The objectives of SG-Mixes, however, it to decorrelate the input-output traffic relationships at an individual node, and the methods employed do not extend to networks of queues.

Source location privacy problem in sensor networks is studied in [34, 35], where phantom routing, which uses a random walk before commencing with regular flooding/single-path routing, was proposed to protect the source location. In [36, 37], Deng proposed randomized routing algorithms and fake message injection to prevent an adversary from locating the network sink based on the observed traffic patterns.

In [11], a distributed anonymity algorithm was introduced that removes fine levels of detail that could compromise the privacy associated with user locations in location-oriented services. In [47], to reduce the probability of tracking users' path via trajectory-based linking continuously collected location samples, a path perturbation algorithm was proposed.

Chapter 4

Traffic Privacy in Wireless Sensor Networks

You can observe a lot just by watching.

Sensor networks are expected to be deployed to perform application-specific sensing and monitoring functions. Some of the unique characteristics of these networks will be their traffic patterns. Traffic in a sensor network tends to flow towards one or more sinks. Further traffic mostly exists when events of interests occur. Data relayed for each type of event dictates the size of the message sent by a sensor node. While the application data contained in the packets may be encrypted to protect it, the context surrounding the creation and transmission of these messages may reveal information to an unauthorized observer of the network. One such context is the size of the message being transmitted by a node. Imagine an adversary observing the traffic in an arbitrary part of the network. Simply observing the message size, of a transmission he intercepts, provides information about the occurrence of a particular event in the network. This is what we define as a violation of traffic privacy in sensor networks.

For example, say a sensor network has been deployed in a war zone to monitor the movement of friendly as well as enemy convoys and troop movements. Such a network may contain a network of motion activated cameras that send information using a network of sensor nodes that also monitor seismic activity. Obviously the packet sizes for these two type of applications are quite distinct simply because of the amount of information that needs to be conveyed is much larger in case of the cameras (either images or video). Even if the data itself is encrypted, the size of the packets or traffic bursts would reveal to an adversary if one or both of the seismic and camera events occurred. That in a real-world means the difference between a stray animal only triggering the camera vs. a convoy of vehicles which would also trigger the seismic sensors. This coupled with reasonable knowledge of the topology (Chapter 3) translates a benign traffic observation into a serious privacy breach.

4.1 Defining Traffic Privacy

Now let us define the problem more generally and formally. Consider a multi-application sensor network. Each application may have several types of events that it monitors. As each type of event is triggered or observed by a node in the network, it will record and/or generate data corresponding to that event and send it in the form of messages towards the sink. Let the set of all possible events in the network be denoted by $E = \{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$ with probabilities of occurrence $P = \{p_1, \ldots, p_n\}$ and message sizes $S = \{s_1, \ldots, s_n\}$. Consider an adversary observing traffic at an opportune point in the network; a location close to the sink would offer the best chance of observing traffic arriving from all parts of the network. The adversary can attempt to infer which event has occurred simply by looking at the size of the message it intercepts over the air. Initially, for simplicity, let us assume that all events are equally likely. If an adversary observes a packet of size s, the traffic privacy can be defined as the uncertainty (entropy) of the adversary's inference about the event given his observation of s:

$$H(\mathcal{E}|s) = -\sum_{s_j=s} \frac{p_j}{\sum\limits_{s_j=s} p_j} log \left(\frac{p_j}{\sum\limits_{s_j=s} p_j} \right) \tag{4.1}$$

A network designer's goal would be to maximize this entropy by altering the sizes of packets transmitted after each event. Let these altered packet sizes be reflected as $S' = \{s'_1, \ldots, s'_n\}$. The problem of providing traffic privacy then becomes:

Maximize $\sum_{j=1}^{n} H(\mathcal{E}_j|s)$ while minimizing the expected communication overhead due to the privacy protection methods: $\sum_{j=1}^{n} p_j(s'_j - s_j)$

There are two broad strategies to improve the privacy in this situation:

- Constant packet size: The key idea is to attempt to make all packets in the network have the same size.
- Randomized packet sizes: This approach relies on randomizing the size of every single packet to create an uncertainty about the type of the packet and underlying event in the mind of an unauthorized eavesdropper.

Let us look at these two strategies in detail.

4.2 Privacy Through Constant Packet Size

One way to hide the type of any packet from the adversary is to make all packets in the network the same size. This can be done by fixing a constant packet size B, then padding all packets smaller than a desired packet size of B bytes with enough arbitrary bits to make the packet size equal to B. If a packet is larger than B bytes, it can be split into several packets of B bytes, with any left over data padded with bits to fill out a last B byte packet. There is a drawback in this method when the packet is larger than B bytes. If an adversary knows that we are breaking our packets into B byte blocks, then he can count the amount N of B byte blocks and know that the original message size was between $(N-1)B$ bytes and NB bytes. Formally we note this as the observation of a *traffic event* of size NB. Therefore, although we have introduced some uncertainty, the adversary is still able to narrow down the message size. To alleviate this problem, these segmented pieces can be sent to different neighboring nodes, thus taking different paths to the sink; or each piece can be sent out after a random amount of delay [48], so that the adversary cannot make out that these pieces belong to the same message. However, random delays require a large outgoing buffer, which may not be available for many sensor nodes. An alternative strategy would employ a value for B' that is larger than the maximum message size, and then the message would be padded to fill out B bytes. Hence we simply transmit a larger B' byte packet.

These methods are good when the variations in the packet sizes, across events, are small. However, a drawback of both of these approaches is that they increase the amount of bandwidth and transmission energy consumed by the sensor nodes.

Intuitively, we can see that to achieve a tradeoff between privacy and packet (and consequently energy) overhead, the network designer would have to choose some packet size less than the maximum. The third set of variables in this calculation are the probabilities of occurrence of different events which contribute the dimension of how likely a certain level of uncertainty is. Let us look at a simple example below to understand this approach to privacy.

Consider four events $\mathcal{E}_1 - \mathcal{E}_4$ that have to be reported by the sensor nodes by packet transmissions. Due to differences in the amount of information that have to be transmitted, we let the packet sizes be variable. In terms of units of the least packet size, let their sizes be 1,2,3 and 4 units respectively. Let the underlying probability of the occurrence of the k^{th} event be p_k. Let us compute the entropies (U) associated with transmitting packets with their true sizes as opposed to transmitting packets by either fragmenting them or adding extra bytes in order to mask their true sizes. Let us also compute the expected overhead (E(V)) incurred when a particular packet size is chosen as the constant packet size. The expected overhead is the sum of the products of overhead for each event and its probability. Figure 4.1 illustrates the overhead calculations for this example.

1. *Packets with true sizes:* Length of a packet indicates the event and hence there is no uncertainty to an adversary who observes the packets and can determine there sizes. Hence $U_0 = 0$. The overhead is $E(V_0) = 0$.

2. *Packets sent by fragmenting them to length of 1 Unit:* In this case event $\mathcal{E}_1$ would generate 1 packet of 1 unit while event $\mathcal{E}_4$ would generate 4 packets of size 1 unit. Even in this case the adversary can be sure about the event by observing the number of packets. Hence $U_1 = 0$ and overhead $E(V_1) = 0$.

3. *Packets sent by fragmenting them to length of 2 Units:* In this case whenever the adversary sees one packet he knows that it is either from $\mathcal{E}_1$ or $\mathcal{E}_2$ and whenever he sees two packets he knows that its from $\mathcal{E}_3$ or $\mathcal{E}_4$. The uncertainty associated with making a decision is,

$$\begin{aligned} U_2 = (p_1 + p_2) & \left[-\frac{p_1}{p_1+p_2} \log\left(\frac{p_1}{p_1+p_2}\right) - \frac{p_2}{p_1+p_2} \log\left(\frac{p_2}{p_1+p_2}\right) \right] \\ + (p_3+p_4) & \left[-\frac{p_3}{p_3+p_4} \log\left(\frac{p_3}{p_3+p_4}\right) - \frac{p_4}{p_3+p_4} \log\left(\frac{p_4}{p_3+p_4}\right) \right] \\ = - & \sum_{k=1}^{4} p_k \log(p_k) + (p_1+p_2)\log(p_1+p_2) + (p_3+p_4)\log(p_3+p_4) \end{aligned} \tag{4.2}$$

The associated overhead is $E(V_2) = 1 \times (p_1 + p_3) = p_1 + p_3$ packet units.

4. *Packets sent by fragmenting them to length of 3 Units:* Similar to the derivation of U_2 we can show that,

$$U_3 = -\sum_{k=1}^{3} p_k \log(p_k) + (p_1 + p_2 + p_3)\log(p_1 + p_2 + p_3). \tag{4.3}$$

The associated overhead is $E(V_3) = 2 \times (p_1 + p_4) + 1 \times p_2 = 2(p_1 + p_4) + p_2$ packet units.

5. *Packets sent by fragmenting them to length of 4 Units:* Similar to the derivations of U_2 and U_3 we can show that,

$$U_4 = \sum_{k=1}^{4} p_k \log(p_k). \tag{4.4}$$

The associated overhead is $E(V_4) = 3p_1 + 2p_2 + p_3$ packet units.

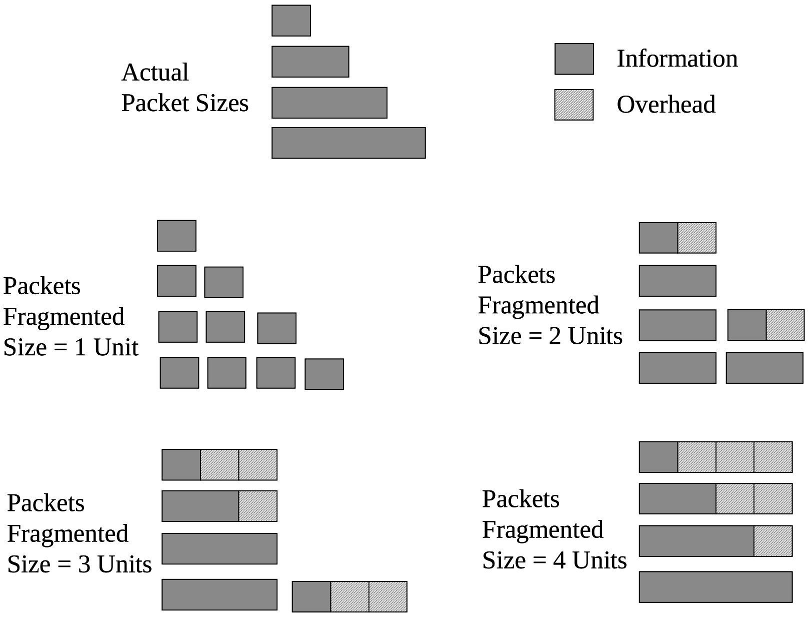


Figure 4.1: Sample overhead calculations in constant packet size strategy.

4.2.1 Formulating traffic privacy and overhead

Let us now derive the general formula for the traffic privacy (entropy) of the network when the constant packet length strategy is in use. Let $E = \{\mathcal{E}_1 \ldots \mathcal{E}_n\}$ be the set of possible events in the network with probabilities of occurrence $P = \{p_1 \ldots p_n\}$ and message sizes $S = \{s_1 \ldots s_n\}$. Let s_c be the constant packet size chosen for the network. Then the new set of packet sizes for the network is $S' = \{s'_1 \ldots s'_n\}$ where $s'_i = s_c * \lceil (s_i/s_c) \rceil$. The entropy and the overhead associated with achieving it can then be calculated as:

$$Entropy\ U = -\sum_{i=1}^{n} p_i \log \left(\frac{p_i}{\sum\limits_{s_i = s_j} p_j} \right) \tag{4.5}$$

$$Expected\ Overhead\ E(V) = \sum_{i=1}^{n} p_i (s'_i - s_i) \tag{4.6}$$

Ideally we would want to have the maximum possible entropy for a given sensor network. However, this entropy comes with the price of having to transmit extra bits and that can be a problem

```
Algorithm: MaxEntropy_ConstSize (In: S, P     Out: Const_size, Entropy)
n = size(P);
for ($idx = 1; idx <= n; idx+ = 1$) do
    b = S(idx);
    curr_lim = b;
    plogp_sum = 0;
    p_sum = 0;
    U(idx) = 0;
    V(idx) = 0;
    for ($i = 1; i <= n; i+ = 1$) do
        if ( S(i) > curr_lim ) then
            curr_lim = curr_lim + b*ceil(S(i)/b);
            U(idx) = U(idx) - plogp_sum + p_sum*$log_2$(p_sum);
            plogp_sum = 0;
            p_sum = 0;

        end
        plogp_sum = plogp_sum + P(i)*$log_2$(P(i));
        p_sum = p_sum + P(i);
        V(idx) = V(idx) + P(i)*(curr_lim - S(i));

    end
    U(idx) = U(idx) - plogp_sum + p_sum*$log_2$(p_sum);
end
Const_size = Smallest $s_i \in$ S s.t. $u_i/v_i$ == MAX($u_i/v_i$) $\forall$ $u_i \in U$ and $v_i \in V$ and
Entropy = $u_i$;
```

Algorithm 3: Maximizing Entropy per bit of overhead with constant packet size

in such networks. Therefore a more appropriate goal would be find a way to get the most efficient entropy improvement possible. We call this metric Entropy Efficiency (EE). Entropy Efficiency is defined as the improvement in entropy per bit of overhead. It is calculated by dividing the entropy improvement due to the padding of packets by the expected overhead in bits due to the padding. Algorithm 3 calculates the *ideal* constant packet size from the set of packet sizes of all events in the network using this metric.

4.3 Privacy Through Randomized Packet Sizes

While the constant packet size strategy may lead to improvement in traffic privacy, it does comes with the constant cost of bits added to each packet. Extra bits to transmit translates to higher energy consumption and longer communication times in the shared-medium, multi-hop sensor networks. In order to reduce the amount of energy consumed and bring down the communication times, we might be willing to allow the adversary to have *some* knowledge of the true packet size or at least its lower bound by never splitting them. One approach to accomplish this, which is motivated by data perturbation methods from data mining privacy [29], is to have each packet append a random amount of bits to the end of the data packet. Since the amount of additional bits is random, it introduces partial uncertainty for the adversary to guess at the packet's original size.

As an example, suppose an unaltered packet for event $\mathcal{E}_1$ is 30 bytes, while an packet for event $\mathcal{E}_2$ is 20 bytes. If we add a random amount of bytes chosen according to a uniform distribution between 0 and 30 bytes, then the altered $\mathcal{E}_1$ packets would range from 30 bytes to 60 bytes, while the altered $\mathcal{E}_2$ packets would range from 20 bytes to 50 bytes. If an adversary observes a 45 byte packet, there is uncertainty in his inference of the underlying event. On the other hand, if an adversary sees a 25 byte packet, then he can definitively conclude that the packet corresponds to event $\mathcal{E}_2$. Let us delve deeper into how such random padding would work for any general network.

4.3.1 Universal random padding range

A simple way to start would be to have the network designer pick a number B for the entire network, to indicate the maximum number of bits that can be padded. For each occurrence of any event a random number of bits (Z) is picked, uniformly, from 0 to B. This is then padded to the original packet size Y and a new packet of size $X = Y + Z$ is sent out. Since the extra bytes are picked uniformly, the expected per packet overhead due to this privacy measure is $B/2$. The network designers goal is to pick a B that achieves $min\ E\{Z\}$ while maximizing the uncertainty in Y given the adversary observing X, i.e. $max\ H(Y|X)$. This uncertainty can be defined as the conditional entropy of Y given X :

$$Given:\ X\ =\ Y + Z\ and\ Z \sim \cup(0 \ldots B-1)$$

$$H(Y|X)\ =\ -\sum_x \sum_y p_{X,Y}(x,y)\log_2 p_Y(y|x) \tag{4.7}$$

$$p_{X,Y}(x,y) = \frac{p_Y(y)}{B},\quad \forall\ 0 \le x-y \le B-1 \tag{4.8}$$

$$p_Y(y|x)\ =\ \frac{p_{XY}(x,y)}{p_X(x)}\ =\ \frac{p_Y(y)/B}{p_X(x)}\quad \forall\ 0 \le x-y \le B-1 \tag{4.9}$$

$$H(Y|X)\ =\ -\sum_x \sum_y \frac{p_Y(y)}{B}\log_2\left(\frac{p_Y(y)/B}{p_X(x)}\right)\quad \forall\ 0 \le x-y \le B-1 \tag{4.10}$$

The entropy of the system can be computed as follows: Let $E = \{\mathcal{E}_1, \ldots, \mathcal{E}_n\}$ be the set of possible events in the network with probabilities of occurrence $P = \{p_1, \ldots, p_n\}$ and message sizes $S = \{s_1, \ldots, s_n\}$. Given that each packet maybe padded with randomly chosen bits from 0 to $B-1$, the new sets of packet sizes and probabilities for the network become:

$$S' = \{s_i^j\}\ \text{ where }\ s_i^j = s_i + j\ \ \forall\ s_i \in S\ \text{ and }\ j = \{0 \ldots B-1\}, \tag{4.11}$$

$$P'\ =\ \left\{p_i'\ =\ p_i.\frac{1}{B}\right\}\ \forall p_i\ \in\ P. \tag{4.12}$$

The entropy of the network and the overhead associated with achieving it can then be calculated as:

$$\text{Entropy } U = -\sum_{p_i' \in P'} p_i' \log\left(\frac{p_i'}{\sum_{s_i'=s_j'} p_j'}\right)\ \forall\ s_i', s_j'\ \in\ S' \tag{4.13}$$

$$\text{Expected Overhead } E(V) = \sum_{p_i^j \in P'} p_i^j(s_i^j - s_i)\ \forall\ s_i^j\ \in\ S'\ and\ s_i\ \in\ S \tag{4.14}$$

We should note that there are conditions on what values of B may be chosen. In the example above, instead of picking a $B = 30$, suppose we picked $B = 5$. Note that the difference between the packets for the 2 events is 10 bytes. Now the packets for event $\mathcal{E}_1$ would range from 30 to 35 bytes and those for event $\mathcal{E}_2$ from 20 to 25 bytes. Obviously there is absolutely no overlap between these two packet size ranges and thereby no privacy achieved. We can see that one would have to pick a $B \geq 10$ to have some probability of uncertainty between the two events. More generally this can be stated as

$$B > \min |s_i - s_j| \quad \forall \quad s_i, s_j \in S. \tag{4.15}$$

4.3.2 Improved random padding ranges

Picking the same B for all events may not always be the best option, especially if the variance in the packet sizes for various events is large. In the first example above, when the packet size for event $\mathcal{E}_1$ is greater than 50 bytes there is not uncertainty as to which message is being transmitted. Therefore, the extra bytes are actually wasted without adding any privacy. In general it makes more sense for the events with larger packet sizes to have smaller ranges and those with smaller packet sizes to have larger ranges to bring about uncertainty in a cost efficient manner.

In light of this, we propose two improvements to calculating the range from which the random bit paddings are chosen:

- Do not add any bits to the largest packet size(s).

- Instead of picking a universal range for the entire set of packet sizes, pick an individually tailored range for each packet size. Let M be the largest packet size in the set; we pick $B \leq M$ as our global range parameter. Then for each packet size s the individual range parameter is:

$$B_s = \begin{cases} B - s & \text{if } s \leq B \\ 0 & \text{Otherwise} \end{cases}$$

With these changes we are being much more conservative and judicious in terms of the maximum number of bits that get padded onto packets, thereby lowering the privacy-related overhead. The conditional entropy of any event given an adversary traffic observation (follows from equation (4.10)) can now be calculated as follows:

$$Given: \ X \ = \ Y + Z \ \ and \ \ Z \sim \cup(0 \ldots B_y - 1) \ \ \forall \ \ y \in Y$$

$$H(Y|X) \ = \ -\sum_x \sum_y \frac{p_Y(y)}{B_y} \log_2 \left(\frac{p_Y(y)/B_y}{p_X(x)} \right) \ \ \forall \ \ 0 \leq x - y \leq B_y - 1 \tag{4.16}$$

Now that each packet with size s_i maybe padded with randomly chosen bits from 0 to $B_i - 1$, the new sets of packet sizes and probabilities for the network is

$$S' = \{s_i^j\} \ \text{ where } \ s_i^j = s_i + j \ \ \forall \ \ s_i \in S \ \text{ and } \ j = \{0 \ldots B_i - 1\} \tag{4.17}$$

$$P' \ = \ \left\{ p_i' \ = \ p_i . \frac{1}{B_i} \right\} \ \ \forall \ \ p_i \ \in \ P. \tag{4.18}$$

The entropy and overhead formulae remain the same as in equations (4.13) and (4.14) respectively.

4.4 A greedy algorithm to maximize privacy

In the preceding sections we discussed ways to enhance the traffic privacy of the system and the resultant equations that can be used to study the tradeoff between entropy or entropy efficiency and overhead for a given choice of parameter such as constant packet size or maximum padding range. However, it would be useful if we can also answer the questions: *If a network is able to tolerate an expected privacy overhead of V bits what is the maximum entropy one can get for the network and* *how precisely to alter the original packet sizes to achieve it ?* In this section we describe a greedy algorithm (algorithm 4) that finds a possibly altered set of event packet sizes S' that would yield the maximum increase in privacy for a given budget of bits to be expended in the form of expected overhead.

We begin by calculating the entropy of the system as is, without making any changes. This serves as the baseline entropy and any changes we make have to provide some positive improvement over the baseline. We then begin the main while loop and divide all the event packet sizes into clusters such that all packet types with same size belong to the same cluster. If the total number of clusters is one then the algorithm stops; there is no need to make any [more] changes because the all the packet sizes are equal and the entropy is at the highest possible value. If there are more than one clusters they are sorted and processed in ascending order of packet size. The algorithm examines one cluster at a time. Every possible subset of the cluster is processed one at a time. Appropriate number of bits (b) are added to the packet sizes in each subset to make the packet sizes equal to the sizes in the next higher cluster. In essence, we are trying to make as many packet sizes equal as possible (to get highest entropy) with minimum number of bits spent. The algorithm then checks the entropy of the system with the modified set of packet sizes and also computes the cost in terms of the expected overhead incurred by adding b bits to this subset. We take this modification of sizes into consideration only if all of the following conditions are met:

- The cost incurred when added to the total cost so far keeps us within the bit budget.
- The change does not increase the highest packet size any further.
- The entropy efficiency gained by this step is higher than any other obtained so far in this round.

This process is repeated for each subset of every cluster except the one with highest packet sizes.

Obviously there is no point in increasing the size of the largest packet sizes since it's not going to add any value in the form of entropy. Having gone through every subset of every viable cluster in this round, we check if any viable set of changes to packet sizes have been found in this round. If yes, then with that as the new set of sizes the loop is repeated. If not, the algorithm exits and returns the current set of packet sizes and corresponding entropy and accumulated overhead.

At each stage the algorithm greedily picks a packet size allocation that gives the best entropy increase per bit for that round or in other words, the highest entropy efficiency. To yield the optimal budget allocation an algorithm would have to go depth first through every possible option at each stage. This can quickly lead to an exponentially large set of choices to be tested and therefore infeasible.

4.5 Concluding remarks

In this chapter, we defined the traffic privacy problem in wireless sensor network and presented a information theoretic formulation of the same. We provided a quantifiable metric to measure the traffic privacy of a network and presented solutions to improve it by using various packet resizing mechanisms. In the end we illustrate a greedy algorithm that computes an allocation of packet paddings that would yield quasi-optimal privacy for a given budget of bits.

This chapter concludes our discussion on the contextual privacy problems in sensor networks. In the next part of this thesis we examine the privacy (and security) challenges in another type of emerging network, namely vehicular ad hoc networks.

```
Algorithm: MaxEntropy_given_Budget (In: S, P, B    Out: S’, H, overhead)
$H_{curr}$ = $calc\_entropy(S, P)$
$S_{curr}$ = S
$rem\_budget = B$
$max\_s = Max(S)$
$overhead = 0$
while (1) do
    k = 0
    Clusters = set{clusters of all $s \in S$ s.t. if $s_i$ == $s_j$ then $s_i$ and $s_j$ are in the same cluster}
    if ( count(Clusters) == 1 )    break;
    Clusters = sort_ascending (Clusters);
    fwd_progress = false
    $U_{max}$ = 0
    $H_{max}$ = $H_{curr}$
    $S_{max}$ = $S_{curr}$
    for each cluster $\in$ Clusters do
        if ( last cluster )    break;
        for (each subset sub $\in$ cluster) do
            Add b bits to each $s \in sub$ to make it equal to the size of the packets in the next higher size cluster.
            Let $S_{temp}$ be the new set of packet sizes for all events.
            $cost_{temp}$ = $\sum_{s \in sub} (b.P_s)$
            $H_{temp}$ = calc_entropy ($S_{temp}$, P)
            if ($cost_{temp} \leq$ rem_budget) && (Max($S_{temp}$) $\leq$ max_s) then
                $U_{temp}$ = $(H_{temp} - H_{curr})/cost_{temp}$
                if ($U_{temp} > U_{max}$ ) then
                    $U_{max}$ = $U_{temp}$
                    $S_{max}$ = $S_{temp}$
                    $H_{max}$ = $H_{temp}$
                    $C_{max}$ = $cost_{temp}$
                    fwd_progress = true
                end
            end
        end
    end
    if ( fwd_progress ) then
        overhead += $C_{max}$
        rem_budget -= $C_{max}$
        $H_{curr}$ = $H_{max}$
        $S_{curr}$ = $S_{max}$
    end
    else    break;
end
S’ = $S_{curr}$
H = $H_{curr}$
return
```

Algorithm 4: Calculate maximum possible entropy H given a budget of B bits

Chapter 5

Privacy in vehicular ad hoc networks

Never answer an anonymous letter.

Vehicular networks are a special case of ad hoc networks and have slightly varying needs. Communication in these networks has to be effective for vehicles moving at high speeds. Vehicles are not constrained by power. They can also be built with high-end computing capabilities. A communication framework for vehicular networks has to provide efficient secure communication between all entities. In this chapter we present a comprehensive security and privacy framework that addresses this problem. There are several different types of communications that can occur in a VANET (see Figure 5.1), as characterized by which entity the vehicle is communicating with:

Vehicle-to-vehicle : This by far is the most critical kind of communication wherein vehicles inform each other of their speeds, location and actions they are taking. These could include time-sensitive actions such as braking, changing lanes, swerving, turning, slowing down etc.

Vehicle-to-BaseStation : Vehicles would use the roadside infrastructure (base stations) to receive traffic updates, road conditions, weather information etc and could in turn provide their own information such as speed and lane number and may be even a projected near-term destination as feedback to predict traffic patterns.

Vehicle-to-Internet : This probably is the least critical kind of communication where passengers in a vehicle are able to connect to the internet using a combination of other vehicles (as a multi-hop network) and the base stations along the road. It is a value adding proposition that would be

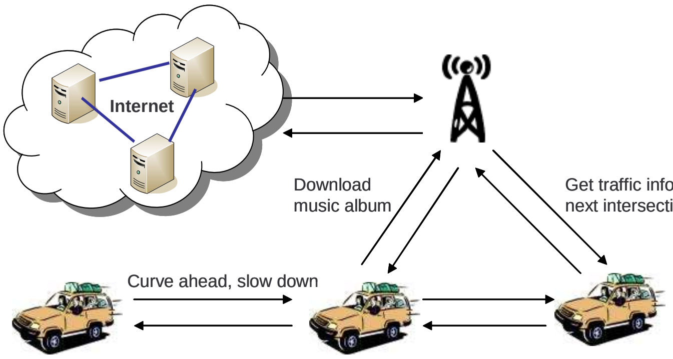


Figure 5.1: Communication in vehicular networks

provided to the consumers possibly as a premium service.

Adversarial models for such communication networks range from mere eavesdropping to malicious data injection to cause accidents. Any security solution has to account for the contention-based opportunistic communication medium, transient association and ad-hoc group formations between vehicles, ease of eavesdropping and disrupting the message and data exchange, high mobility, and the acute need for privacy in these networks.

Recent works have addressed some of the security and privacy issues in vehicular networks [49–51]. However, as we shall elaborate later in this chapter, these solutions are rather inflexible when it comes to balancing the need for privacy (pseudonymity) with authentication and non-repudiation.

Our framework, exploits the inherent properties of identity based cryptography, such as the use of identities (pseudonyms in our case) as public keys and implicit authentication, to provide a more versatile solution. It provides greater flexibility to users regarding their privacy needs with marginal additions to the communication and storage costs of the infrastructure. The contributions of our work are as follows:

- We propose a security framework for vehicular networks using Identity-Based Cryptography

(IBC). Our framework provides for authentication, confidentiality, message integrity, non-repudiation and pseudonymity.

- We present a pseudonym generation mechanism that leverages the unique characteristics of vehicular networks such as the presence of roadside base-stations and vehicle mobility and interaction patterns. It allows for user-controlled levels of privacy (pseudonymity) and yet provides non-repudiation because [only] a Trusted Arbiter can reconstruct the true identity of a vehicle from its pseudonym. Our mechanism does not require the storage of any pseudonym related information either at roadside base-stations or at the Trusted Arbiter nor does it require any per-pseudonym communication between the two.

- Our framework allows for customizable trust and privacy settings where the vehicle owner can change its pseudonyms at a frequency based on his privacy requirements and also use the age of a pseudonym of other vehicles as a measure for varying levels of trusted interactions. This can serve as a building block for other application level protocols and services that require customizable trust thresholds.

The rest of the chapter is organized as follows : in Section 5.1 we outline the requirements of a security framework for vehicular networks, in Section 5.2 we discuss the cryptographic background on identity-based cryptosystems; in Section 5.3 we present the details of our joint security/privacy framework; Section 5.4 analyzes the performance of the framework and Section 5.5 describes its advantages.

5.1 Privacy and Security of Vehicular Networks

A security framework for VANETs must address the usual security objectives. However, since VANETs will be involved with one of the most prevalent social actions in modern society, i.e. driving, these requirements become more pronounced. In particular, a security framework for VANETs should address the following requirements:

Authentication: A vehicle must be able to prove its identity to a base station or another vehicle (the same holds for the base station). Here we refer to the term identity loosely. There is no need for a vehicle to ascertain the true identity of the party its talking to as long as some trusted authority has checked that identity and this fact can be verified from the pseudonym being used. We do need a mechanism that prevents masquerade attacks.

Confidentiality: We need the ability for parties to interact with each other securely without having their conversation snooped on. While an adversary can certainly capture the packets over the air, she should not be able to decrypt them.

Non-repudiation: As mentioned earlier, we may not be interested in knowing the true identities of the parties we are interacting with as long as we have proof they are legitimate identities. However given that adversaries may inject false information into the network, we need the ability to unequivocally link communication back to its originator with the help of a trusted arbitrator. This way parties cannot deny their part in a particular exchange and can be held liable for their actions. This property will serve as a deterrent to parties from injecting malicious data into the network.

Privacy: An adversary should not be able to track a vehicle's activities purely based on its communications. A stronger requirement is that the system itself should not be able to track individual vehicles solely based on the pattern of interaction of a vehicle with different base stations. Such an ability would be perceived as highly intrusive by drivers and would act as a deterrent to widespread user participation in these networks.

Message Integrity: Vehicles should be able to detect messages that have been corrupted during transmission or injected by a malicious adversary. Any security framework should have the ability to rapidly authenticate messages and verify that they are indeed from the source they claim to be from. This is most significant when it comes to traffic safety messages. These are messages which inform neighbors of a vehicle that it is braking, changing lanes, etc. Because these safety messages have to be acted upon as soon as possible, we need to make sure that they reach as many neighbors as quickly as possible and their source and integrity can be verified expediently.

Resilience: VANETs, like other wireless systems, will be susceptible to a variety of denial of service (DoS) attacks, ranging from resource exhaustion attacks to jamming. Although it might be difficult to prevent jamming attacks [52] and adaptive radio techniques might be employed to mitigate this threat, various other forms of DoS attacks that target buffers and cause a wastage of communication and computational resources (whether at the base-station or at another vehicle) can be prevented by appropriate cryptographic mechanisms. As we will see, some of the related work requires vehicles to buffer messages until they reach a base-station to then decrypt them. This makes the vehicles susceptible to being flooded with junk messages, overwhelming their buffers and later wasting their precious communication and computational resources to transmit these messages to the base-station for decryption and authentication. This attack is certainly preventable by giving vehicles the ability to instantly authenticate the data transmission they receive.

5.2 Identity-Based Cryptography

We now provide a brief overview of identity-based cryptography that serves to provide a context and a common frame of reference for our proposed VANET security framework. In 1984, Adi Shamir first proposed the idea of an identity-based cryptosystem [53] in which arbitrary strings can act as public keys. However, it was only in 2001 that the first practical identity-based encryption (IBE) scheme was produced by Boneh and Franklin [54]. Their scheme uses a non-degenerate, bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, where $\mathbb{G}_1$ and $\mathbb{G}_2$ are cyclic groups of order p for some large prime p. In particular this map satisfies the following property :

$$\hat{e}(aP, bQ) \ = \ \hat{e}(P, Q)^{ab} \ \forall \ P, Q \in \mathbb{G}_1 \ \ and \ \ a, b \in \mathbb{Z}_p.$$

Weil and Tate pairings on elliptic curves are two fast and efficient ways of constructing such bilinear maps. Identity based encryption mechanisms broadly involve the following operations:

Setup : A Trusted Authority (TA) chooses an elliptic curve, a random secret s and a point P on the curve. It distributes the public parameters: s and sP to all the participants in the system.

Extract : In this phase the TA extracts the private key for a given identity (public key). Say Bob uses ID_{Bob} as his public key, he can obtain the corresponding private key $s.ID_{Bob}$ from the TA. Note that only the TA can compute this key since it alone knows the secret s.

Encrypt : To send an encrypted message to Bob, Alice can compute the encryption key k by picking a random r and computing $k = \hat{e}(rID_{Bob}, sP)$. She then sends the encrypted message $Encrypt(k, Message)$ and rP to Bob.

Decrypt : Because of the bilinearity property of the system Bob can compute the key k independently using : $k = \hat{e}(sID_{Bob}, rP)$

As can be seen from the steps above that IBC eliminates the need for certificates with its implicit authentication. The sender does not have the burden of verifying an identity of the recipient because the recipient has to authenticate himself to a common trusted authority in order to obtain the private key corresponding to the claimed identity. Moreover identities, which serve as public keys can be specially constructed strings. We exploit these properties to generate the unforgeable pseudonyms that serve as time-varying identities of the vehicles.

5.2.1 Identity-based signcryption

In addition to encryption we need the ability to provide message integrity and non-repudiation in a cost-effective manner. In order to achieve this with relatively meager computational requirements, we have chosen to employ identity-based signcryption. In 1997, Zheng [55] first introduced a cryptographic primitive called, *Digital Signcryption*, where he combined encryption and digital signature schemes to do the two operations at a much lower cost as compared to doing signature-followed-by-encryption. While his scheme was based on ElGamal signatures and encryption, similar schemes have been developed for identity-based cryptosystems. We chose a fairly recent identity-based signcryption scheme by Chen and Malone-Lee [56] for our framework. For our purposes this is the most efficient scheme, we came across, both in terms of computational cost savings and reduced size of cryptograms [56], as compared to encrypt-then-sign methods. It works as shown below

Setup : Create system parameters $\mathbb{G}_1, \mathbb{G}_2, \hat{e}, q$ and a random generator $P \in \mathbb{G}_1$. Let k_0, k_1 and n be the number of bits required to represent an element of $\mathbb{G}_1$, an identity and the message respectively. The following hash functions also need to be established

$H_0 : \{0,1\}^{k_1} \rightarrow \mathbb{G}_1$

$H_1 : \{0,1\}^{k_0+n} \rightarrow \mathbb{Z}_q^*$

$H_2 : \mathbb{G}_2 \rightarrow \{0,1\}^{k_0+k_1+n}$

Pick a random $s \in \mathbb{Z}_q^*$ and compute $P_{pub} = sP$

Extract : Compute $Q_{ID} = H_0(ID)$ and return private key $d_{ID} = sQ_{ID}$

Sign : Consider a plaintext message $M \in \mathcal{M}$ being encrypted by Alice with public identity ID_A and private key d_A.

1. Pick random $\sigma \in \mathbb{Z}_q^*$ and compute $U = \sigma Q_{ID_A}$

2. Compute $h_1 = H_1(U \| M)$ and $V = (\sigma + h_1)d_A$

3. (U, V) is the signature. Send $\langle M, \sigma, U, V \rangle$ to the Encrypt stage.

Encrypt : Using the output $\langle M, \sigma, U, V \rangle$ Alice will encrypt the signed message M, destined for Bob (ID_B) as follows,

1. Compute $Q_{ID_B} = H_0(ID_B)$ and $g = \hat{e}(\sigma d_A, Q_{ID_B})$

2. Compute $W = H_2(g) \oplus (V \| ID_A \| M)$ and set ciphertext $C = \langle U, W \rangle$

Decrypt : Bob with public key ID_B and private key d_B will decrypt $C = \langle U, W \rangle$ as follows

1. Compute $g = \hat{e}(U, d_B)$

2. Compute $V \| ID_A \| M = W \oplus H_2(g)$

3. We now have the message and (U, V) is Alice's signature. Send $\langle M, U, V, ID_A \rangle$ to the Verify stage.

Verify :

1. Compute $Q_{ID_A} = H_0(ID_A)$ and $h_1 = H_1(U \| M)$

2. If $\hat{e}(V, P) = \hat{e}(P_{pub}, U + h_1 Q_{ID_A})$ return *true* else return *false*.

This is the scheme we use to build our security framework. Signcryption produces significantly more compact cryptograms than those produced by encrypt-then-sign schemes [56]. In Section 5.4.3 we illustrate why this is an especially important benefit in vehicular networks. Moreover this particular signcryption scheme saves one Tate pairing in the decrypt/verify process as compared to other signcryption schemes developed before it and one multiplication in $\mathbb{G}_1$ as compared to encrypt-then-sign schemes [56]. As we will see in Section 5.4.1 these are the two costliest computations in IBC.

5.3 An identity-based security/privacy framework

As described in Section 5.1, we want a vehicular network to have authentication, confidentiality, non-repudiation, privacy, data integrity and resilience. Vehicles and base-stations should be able to authenticate themselves and at the same time use disposable pseudonyms for vehicles so that their activities and communications are not tracked by parties that are eavesdropping on them. We also need to make certain that there is a verifiable trail between the pseudonyms and the real identities of the vehicle and that only a common, Trusted Arbiter(TA) is able to verify that trail in case of a dispute. Identity-based cryptography (IBC) lends itself nicely to help solve these problems.

In the discussion that follows, we shall use the following notations:

- ID_v, ID_I : Identities of the vehicle and base-station respectively (base-stations have capital letter subscripts). In case of a vehicle, we use the notation ID_v^i to refer to i^{th} pseudonym. Identifiers like ID_a, ID_b will be used to indicate two vehicles a and b respectively.

- d_v, d_I : Secret key corresponding to ID_v and ID_I respectively.

- K_I : Shared secret key assigned to the base-station I, by Trusted Authority (TA). This key is used with a symmetric key algorithm to generate the vehicle pseudonym as explained later.

- TS_i : Timestamp at time *i*.

- K_{pub}^v, K_{pvt}^v : Public and private keys assigned to a vehicle by TA as part of the certificate issued to them. This certificate contains the vehicle's true identity.

- $sigEncrypt$ (sigencryption) and $sigDecrypt$ (sigdecryption) refer to identity-based operations while $rsaEncrypt$, $rsaDecrypt$, $rsaSign$ and $rsaVerify$ refer to operations that use the RSA algorithm. In some places we breakup $sigEncrypt$ and $sigDecrypt$ to its sub-functions $Sign$, $Encrypt$, $Decrypt$ and $Verify$. Additionally, we use $aesEncrypt$ and

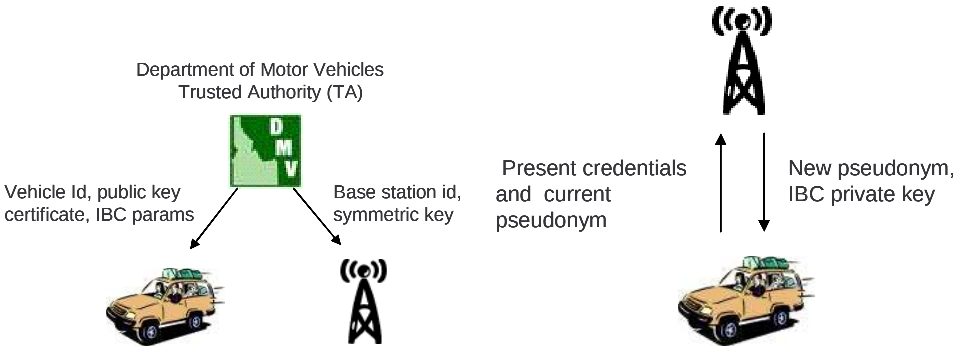
$aesDecrypt$ to denote symmetric cipher operations using the AES cipher.

- $||$ is the concatenation operation.

5.3.1 Framework Description

Below, we describe our identity-based security framework that provides authentication and confidentiality using the signcryption scheme described in Section 5.2. Additionally, the framework scales well to allow user-controlled privacy. Vehicles can create as many pseudonyms as they want without adding any extra storage requirements. Finally, the system guarantees non-repudiation in front of a trusted arbiter by allowing the arbiter to trace back the pseudonym to the permanent, unique vehicle identity.

Each vehicle and base-station has a unique identifier ID_{id}. These identifiers include the designation of the entity as a vehicle or base-station; e.g. $ID_v = (vehicle||identifier)$. We envision that these identifiers can be certified at regular periods by a Trusted Authority (TA) that is trusted by all parties. For example, a natural strategy would be to have the identifiers certified annually. If any certificate is revoked, the TA notifies all the base-stations in the system. Since renewal of credentials is annual, each base-station only has to store Certificate Revocation List (CRL) entries that are less than a year old. Vehicles never have to download any CRLs and this is a big savings in communication costs. We also note that, unlike many other wireless network scenarios, VANETs are likely to be characterized by entities that are not power-limited. Rather, both vehicles and base-stations will have the ability to perform more intensive computations. Already, the current generation of automobiles consist of embedded systems with powerful processing capabilities. It is therefore quite reasonable to assume that entities involved in a VANET will be able to perform operations like Tate pairings.



(a) Setup

(b) Pseudonym generation

Figure 5.2: Setup and Pseudonym generation in VANETs

Setup phase: The Trusted Authority (TA) conducts the setup phase of the identity-based cryptosystem as described in Section 5.2 and computes the relevant system parameters (*params*) and the master secret s. Both of these are then distributed to all the base-stations in the system (see Figure 5.2). Additionally the TA generates a random secret key K_I for each base-station I and distributes it to that base-station. The TA keeps a copy of this key in its database to help in future arbitration proceedings, as we will see later in this section. The TA provides each vehicle with its unique vehicle identifier (ID_v), public key certificate certifying this identifier and including a public and private key pair (Pub_v and Pvt_v) generated using classical algorithms such as RSA. Additionally each vehicle is provided with all the *public* system parameters (params) of the identity-based cryptosystem.

Pseudonym generation : When a vehicle needs to get a new pseudonym it engages a base-station as follows:

$$
\begin{aligned}
ID_v^i &: M = \langle Cert_v, TS_j, ID_v^i, rsaSign_{K_{pvt}^v}(ID_I \| ID_v^i)\rangle \\
ID_v^i \rightarrow ID_I &: C = sigEncrypt_{d_v^i}(ID_I, M) \\
ID_I &: M = \langle Cert_v, TS_j, ID_v^i, U\rangle = sigDecrypt_{d_I}(C) \\
&\quad rsaVerify_{K_{pub}^v}(U, ID_I \| ID_v^i) \\
&\quad T = aesEncrypt_{K_I}(ID_v \| TS_{j+1})) \\
&\quad ID_v^{i+1} = \langle vehicle \| T \| ID_I \| TS_{j+1}\rangle \\
&\quad d_v^{i+1} = Extract(ID_v^{i+1}) \\
ID_I \rightarrow ID_v^i &: rsaEncrypt_{K_{pub}^v}(ID_v^{i+1} \| d_v^{i+1} \| TS_j)
\end{aligned}
$$

Since we assume that base-stations have up-to-date CRLs, they will only issue a new pseudonym if the vehicle's credentials have not been revoked.

Secure communication : When it comes to general communication, be it between vehicles or vehicle to base-station, our system provides an implicit credential in the form of the pseudonym. The pseudonym includes a time-stamp indicating the last time some infrastructure point validated the credentials of a vehicle. Each vehicle could set its trust threshold as per the user's choice, in deciding how old pseudonyms they want to trust. Once that choice is made, we can simply validate the identity-based signature on the message to verify that the vehicle using the pseudonym actually has the private key corresponding to it. We emphasize again that the private key could only have been generated by a base-station (or the TA) who have the master secret s. Consider two vehicles with pseudonyms ID_a and ID_b, exchanging a message m

$$
\begin{aligned}
ID_a &: M = ID_a \| ID_b \| m \\
&: M' = \langle M, \sigma, U, V \rangle = Sign_{d_a}(ID_a, M) \\
&: \langle U, W \rangle = Encrypt_{d_a}(ID_b, M') \\
ID_a \rightarrow ID_b &: C = \langle U, W \rangle \\
ID_b &: \langle M, U, V, ID_a \rangle = Decrypt_{d_b}(U, W) \\
&: \langle U, V \rangle \text{ is } ID_a\text{'s signature on message } M \\
&: \text{If } Verify(M, U, V, ID_a) == true, \text{ accept } M
\end{aligned}
$$

Non-repudiation : In case of an accident or some other general dispute involving vehicles one can try to locate the cause of the incident based on the messages exchanged between vehicles. Vehicles can log messages into some-kind of a black-box like device and turn these messages over to an arbiter. We assume for simplicity that the arbiter is the same as the systemwide Trusted Authority (TA) and has access to the secret key database (containing secret keys of the base-stations). Suppose vehicle ID_b hands over a message M and corresponding signature $\langle U, W \rangle$ stating it was sent by vehicle pseudonym ID_a^i to pseudonym ID_b^i. The arbiter will validate if the message indeed was created and signed by ID_a^i, intended for ID_b^i and then will decipher as to which real vehicle ID's these pseudonyms belong to. This mechanism works as follows

1. $M = ID_a^i \| ID_b^i \| m$
2. Check that ID_a^i and ID_b^i are in M
3. If $Verify(M, U, V, ID_a^i) == true$, continue
4. We know $ID_a^i = \langle vehicle \| T \| ID_I \| TS_{j+1} \rangle$
5. $K_I = KeyLookup(ID_I)$
6. $ID = \langle ID_a \| TS_{j+1} \rangle = aesDecrypt_{K_I}(T)$
7. Check that ID contains the same TS_{j+1} as in ID_a^i
8. ID_a is the real identity of the sender.
9. Repeat steps [4..8] with ID_b^i to get recipient.

The advantage of this scheme is that no special storage is required in either the vehicles or the infrastructure for each pseudonym. The message M containing the source and destination pseudonyms and signature are the only things that need to be stored to settle any disputes. Further, the original identities of the vehicles can be re-created only by a trusted arbiter with valid legal cause for such action.

5.4 Performance Analysis

Normally, cryptosystems based on elliptic curves enjoy a keysize advantage over equivalent asymmetric cryptosytems that utilize conventional integer-based operations. In particular, this is due to the fact that the discrete logarithm problem over elliptic curves is computationally much harder than the classical discrete logarithm problem. However, identity-based cryptography (IBC) is based on supersingular elliptic curves [13], where the discrete log computation can be reduced to the classical discrete log problem. This is not a deterrent for using IBC for VANETs, however, as our objective is not to achieve the computational advantages of elliptic curves, but rather to exploit the unique structural properties of IBC. In particular, it is straight-forward to use larger key sizes (e.g. comparable to conventional RSA cryptography), while having desirable levels of security and the unique advantages of IBC.

5.4.1 Cost of Computing

In order to explore the computational costs associated with our proposed framework, we used the MIRACL [57] C library implementation of Tate pairings and other cryptographic operations discussed in Section 5.2. We measured the computational costs on a 2.8 GHz pentium machine. These operations are CPU intensive and we argue that, given the continuing decline in the cost of CPUs, it is not unreasonable to expect such CPU capabilities to be cost-feasible in future base-stations and even on-board vehicles. As we will see, later in this section, the price we pay in computing cost for IBC operations, buys us communication efficiency in terms of low security overhead. The results in Figure 5.3(a), show the time taken for computing various cryptographic operations involved in our framework. As we can see Tate pairing is the costliest operation in the entire set. However, we note that as part of the overall system, the $\approx 20ms$ time taken for Tate pairing is acceptable for vehicular networks, where we do not have computationally constrained devices. Since the higher level functions in Figure 5.3(b) are constructed using the the above set of operations we can obtain the approximate time needed to execute them.

5.4.2 Service Latency for pseudonym generation

Another important factor to consider, from performance point of view, is how effectively a single base-station can handle requests (for pseudonym generation) from passing vehicles. In North America vehicular networks will use the DSRC standard [58] which uses the 5.9 GHz licensed spectrum.

	Time (ms)
H_0	0.053
H_1	0.012
H_2	0.134
$s.Q_{ID}$	10.538
$\hat{e}(a, b)$	20.5

(a)

	Time (ms)
$Extract$	10.591
$Sign$	10.550
$Encrypt$	20.687
$Decrypt$	20.634
$Verify$	41.065

(b)

Figure 5.3: Computation time in milliseconds for various cryptographic operations. Elliptic Curve: $y^2 = x^3 + x \; mod \, p$

The medium access control (MAC) protocol of this standard is being defined as part of the 802.11p specifications. The 802.11p MAC is based on the widely used 802.11a MAC. Hence we tested the feasibility of obtaining these periodic pseudonyms from a base-station using wireless nodes communicating using 802.11a wireless cards. We ran experiments measuring the service times for the requests made by vehicles for key generation to a base-station.

The experiments were run on an experimental wireless research testbed called ORBIT [59]. ORBIT is an open access wireless testbed designed to conduct repeatable wireless experiments. Each node has 2 wireless cards which can be used in 802.11 a/b/g mode, in addition to the 3 ethernet interfaces meant for data and control flow. To gather our measurements we used the ORBIT Measurement Library (OML) [60], a distributed measurement collection framework available with ORBIT.

Figure 5.4 shows the cumulative distribution of service latency times for a vehicle across 10,000 requests. The x-axis shows the service latency which includes communication time and the time for pseudonym generation on the base-station.

Consider that a vehicle can communicate with a base-station upto 300m away, which is the maximum outdoor range of commonly available 802.11a cards today. Assume average vehicle speed of 100 km/h (27.7 m/s). Given this, a vehicle stays within communicating range of a base-station for a best-case time of $2 * 300/27.7 \approx 21.6$ seconds.

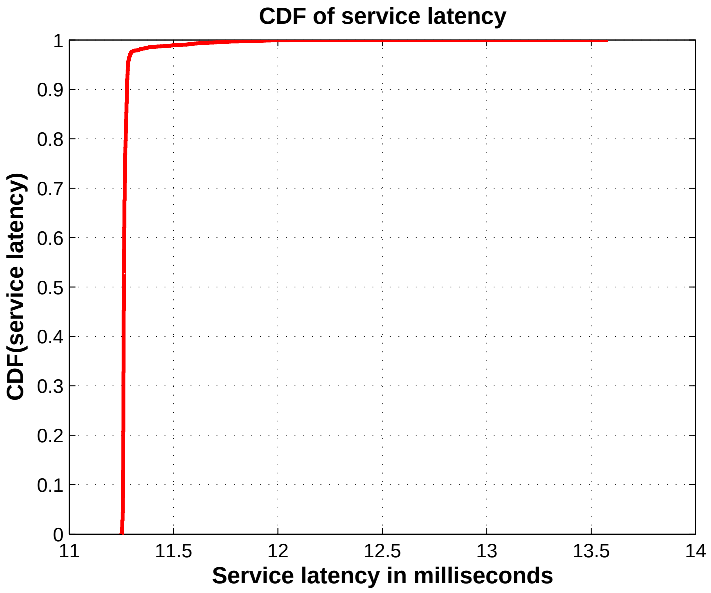


Figure 5.4: Cumulative Distribution Function of service latency for vehicle to obtain pseudonym from a roadside base-station over 802.11a network

As can be seen from Figure 5.4, most of the service times are within the 11 to 11.5 ms range, which is certainly a good time, considering the $\approx$ 21.6 seconds a vehicle will have to complete this task. Assume that the length of a mid-size family sedan in the United States is about 5 meters and let us consider an 8-lane highway (4 lanes on each side), with the base-station on the median and a distance of $\approx$ 30 meters between cars. At the 100 km/h speeds we are assuming, the $30m$ vehicle separation, is a rather strong assumption. The recommended distance, using the *2-second rule*[1], of the department of motor vehicles is 55m. So in a single lane we have $\approx$ 17 vehicles in the 600m diameter that a base-station covers. That translates to a total of 136 vehicles in the 8 lanes and worst case service time of $11.5ms * 136 \approx 1.564$ seconds, assuming all requests are handled sequentially. Note that this is the absolute worst case scenario considering only mid-size sedans, high-density, high-speed traffic and sequential processing of requests and is still significantly below

[1] The rule asks that drivers maintain as much distance between cars, as the car would travel in 2 seconds, at the speed at which it is traveling.

the 21.6 second window the vehicles have.

5.4.3 Communication costs

Communication costs of messaging in VANETs are an important consideration towards their smooth and efficient operation. The CSMA/CA nature of the MAC (Medium Access Control) layer used in VANETs means that the throughput of the network suffers as the payload size in the frame and the number of vehicles contending for the channel goes up. Therefore it is imperative to keep a tight constraint on the added message overhead of the security framework. This assumes more significance when it comes to safety messages, which have to be sent out as quickly as possible and therefore need to be short. Therefore we analyze the overhead of our security framework vs. Public Key CryptoSystem (PKCS) using RSA and ECDSA (elliptic curve based) certificates. The metric chosen for this comparison is frame delay. Frame delay is defined as the time elapsed between a frame reaching the head of the MAC queue (ready to be sent out) and the completion of a successful transmission. Two factors that influence this metric most are the number of contending nodes and the size of the frame payload. Since the set of neighbors a vehicle has in a VANET is continually changing, we have to send out each safety message along with the identifying credentials of the sender and signature computed on the safety message. In terms of traditional PKCS it means the public key certificate of the vehicle would be sent out along with the message and its signature. Assuming a security strength equivalent to 1024 bits of RSA public key, the security overheads can be calculated as follows:

- **Message Overhead of RSA certificates**:

  Size of Signature (Sigsize) = 128 bytes (1024 bits).

  Size of identity information (Idsize) = size of RSA certificate $\approx$ 1200 bytes.[2]

[2] Approx. total size of a X.509 certificate with 1024 bit public key.

Total Overhead = sigsize + Idsize $\approx$ 1328 bytes.

- **Message Overhead of ECDSA certificates**:

  Size of Signature (Sigsize) = 24 bytes (192 bits).

  Size of identity information (Idsize) = size of ECDSA certificate $\approx$ 1016 bytes.[3]

  Total Overhead = sigsize + Idsize $\approx$ 1040 bytes.

- **Message Overhead of our framework**:

  Sigsize = 128 bytes (1024 bits).

  Idsize = 24 bytes (128 bit AES encrypted pseudoId + 32 bit Base-station-ID + 32 bit timestamp)

  Total Overhead = 152 bytes.

We use the delay analysis model derived in [61] to study how the frame delay varies with number of contending nodes for the three frame sizes. Since safety messages are more likely to be broadcast to all the vehicles in the neighborhood, we consider the *basic access* method of communication without RTS/CTS. In the *basic access* method, the access to the medium is regulated using an *InterFrame Space* (IFS) time period and a binary exponential backoff (BEB) algorithm for collision avoidance. The time after an IFS interval is divided into *slots* and nodes are allowed to transmit only at the beginning of a slot. After the channel has been idle for DCF-IFS (DIFS) interval, each node uses the BEB algorithm to compute the appropriate number of slots to wait for before sensing the channel again. Average frame delay (E[D]) is the product of average number of slots required

[3] Assuming same amount of common identification information as in the case of RSA

for successful transmission (E[X]) and average length of slot time (E[S]).

$$E[D] = E[X] \cdot E[S]$$

$$E[S] = (1 - P_{tr}) \cdot \sigma + P_{tr} \cdot P_s \cdot T_s + P_{tr} \cdot (1 - P_s) \cdot T_c$$

$$E[X] = \frac{(1 - 2p) \cdot (W + 1) + pW \cdot (1 - (2p)^m)}{2 \cdot (1 - 2p) \cdot (1 - p)}$$

where

P_{tr} = Probability of at least one transmission in a slot.

P_s = Probability that the transmission is successful.

T_s = Average time the medium is sensed busy due to successful transmission

T_c = Average time the medium is sensed busy during a collision

σ = Duration of empty slot

p = Collision probability

m = Maximum backoff stage

The details for calculating these values can be found in [61]. We modeled a basic access 802.11a network using the appropriate values for parameters, such as header sizes, contention window, interframe intervals, etc. The data rate used was 12 Mbps since DSRC safety messages will be sent at that rate.

Figure 5.5 shows the comparative growth in frame delay as the number of participating nodes increases from left to right. We can see that beyond approximately 10 nodes, the frame delay for a system using RSA certificates (and thereby largest payload) grows significantly faster. The frame delay for our framework always fares substantially better than the other two. This result clearly underscores the advantage of our framework in keeping the security overhead low.

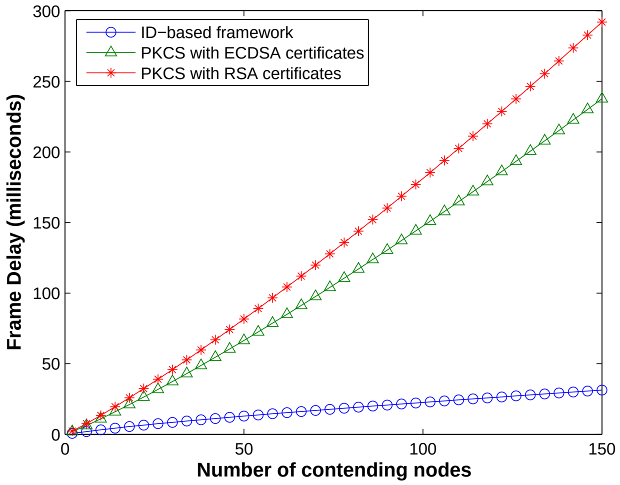


Figure 5.5: Frame delay characteristics in 802.11a network for different payload sizes corresponding to the security overheads of our framework and frameworks using RSA and ECDSA certificates.

5.5 Advantages of the identity-based security/privacy framework

1. **User-controlled privacy levels:** The biggest advantage of our framework comes from the efficiency of providing user-controlled privacy by allowing users to obtain pseudonyms as and when desired. Others have suggested ways to provide the same that are not as flexible or efficient. Proponents of traditional PKI in VANETs suggest pre-computed set of public key certificates being given to vehicles [51]. Every year when a vehicle goes in for inspection it can be given 365 PKI certificates which it can use one per day. This provides limited pseudonymity by mandating one pseudonym for one day. This has several disadvantages. Allowing users to pick and choose the number of pseudonyms they want, makes the system unnecessarily complex and further forces users to anticipate their privacy requirements for a significant period of time into the future and obtain keys accordingly. If these requirements change over time, the user has to go back to the certification authority to obtain a new set of pseudonyms. Adversaries may follow target vehicles during the course of the day and

eavesdrop on their traffic to learn about the interaction of the vehicle and its passengers with their surroundings. Even if the traffic is encrypted, the adversary can determine which entities a vehicle has corresponded with during the time period, if its pseudonym is unchanged during the day.

A user may desire varying levels of privacy on different days or circumstances. Our system allows a vehicle to change it's pseudonym any number of times, at any given time, by requesting a new pseudonym from the base station. The base-station can provide a new pseudonym upon request to any vehicle after verifying its credentials. Note that this mechanism does not require the base station to store the old pseudonyms given to the vehicles, making the design very elegant and scalable at the same time.

2. **Scalability:** In case public key certificates are used for authentication, when the credentials of any vehicle are revoked, every base-station and vehicle in the system has to be notified about it. This generates excessive traffic between base-stations and vehicles just for downloading certificate revocation lists. Alternatively, one could argue that base-stations can act as Certificate Authorities and generate new certificates for users on-demand. The problem with this mechanism is that it doesn't scale. To provide for successful arbitration in the event of a dispute, the arbiter would have to have knowledge of every single pseudonym that is issued by every base-station. So a base-station has to communicate to the arbiter, the public-key/private-key pair for every certificate it issues. Moreover the arbiter has to now store these pairs for a long time (years) in order to provide reliable arbitration. The number of such pairs is proportional to the number of vehicles in the system multiplied by the number of pseudonyms they request.

   In our framework, the pseudonym contains an encrypted version of the vehicle-id, the id of the base-station that created it and a timestamp. The original vehicle-id can be recovered by an arbiter from the pseudonym itself. The arbiter does not need to have information about prior pseudonyms used by this vehicle. We assume that the arbiter has access to the secret keys of every base-station deployed in the system, which is a very reasonable assumption. Since we use symmetric cryptography here, the size of the keys can be small (128 bits). The total number of such keys depends on the number of base-stations in the system and **not** on the number of vehicles and their privacy needs, as required by the PKI approaches discussed previously.

3. **Flexible security and trust:** The pseudonym assigned to a vehicle consists of a timestamp as part of the pseudonym. This timestamp indicates the time when the pseudonym was assigned to the vehicle by the base station. Hence, a more recent timestamp ensures that the vehicle's credentials were not revoked by the TA until the time indicated on the timestamp. Vehicles can use this information to tune their trust levels. For example, a vehicle can place higher levels of trust on vehicles with recently obtained pseudonyms, as opposed to vehicles with older pseudonyms. This feature can be used as a building block for trust based routing protocols and reputation-based ad-hoc network formation and applications. Some researchers have proposed the classification of messages between vehicles into safety and non-safety messages [51]. This framework can be nicely used to provide such demarcation, where applications designed for safety messaging can use higher trust thresholds imposing more stricter requirements, while non-safety messaging can be achieved using less restrictive trust thresholds but exploiting other properties of the network.

4. **Building block for securing VANET applications:** Our framework can be used to secure existing vehicular network protocols and applications. One such example is the Vehicular

Information Transfer Protocol (VITP) proposed by Dikaiakos et. al [62]. This is an application layer protocol where a vehicle can query other vehicles on the road for information. It uses a pull based approach for queries and a push-based approach for distributing alerts. Their protocol description does not address any security issues. Our framework can be used to make this protocol secure. For example, the queries can be sent out containing a vehicle's pseudonym and responding vehicles simply encrypt their responses using this pseudonym. That way the communication is safe from being snooped on by intermediate nodes being used for forwarding.

5.6 Related Work

Raya et al. [63] discuss the various security aspects of inter-vehicular communication. The attacks on vehicular networks discussed in their work include sending bogus information, cheating, identity disclosure and disrupting network operation. They propose the use of electronic license plates for vehicle identification and secure communication using the public key infrastructure. In [51] they extend these ideas and detail a security architecture for vehicular networks using PKI. They discuss issues of key management, anonymity and DoS resilience and the requirements that PKI scheme should meet to be feasible. They propose anonymization of vehicles by pre-loading a vehicle with temporary public/private key pairs and respective certificates. This has two disadvantages. The user needs to predetermine the frequency with which he will change temporary identifiers. The TA needs to store all the temporary keys belonging to all vehicles to resolve disputes, leading to unreasonable demands on the TA. Hubaux et al. [50] outline the privacy issues with the use of electronic license plates and propose secure positioning using verifiable multi-lateration techniques. Parno et al. [64] also propose the use of public key infrastructure (PKI). Anonymous identifiers are provided by base stations running anonymization service. The service verifies the original certificate and provides a new temporary certificate. This approach allows dynamic renewal of anonymous identifiers. However, non-repudiation poses unrealistic storage and communication demands on TA and base stations because the temporary certificates have to be conveyed to the TA in order for it to perform any future audits or arbitration. In general traditional PKI approaches need vehicles to exchange certificates as part of communication with other vehicles, putting a heavy demand on the communication medium as we showed. Additionally, vehicles need to also download the latest certificate revocation lists. Our approach does not use certificates for inter-vehicular communication making the communication faster. [65], [66] and [67] also outline the security and privacy problems that vehicular networks will face.

Choi et al. [49] address the need to balance auditability and privacy requirements in VANETs. They propose the use of symmetric key cryptography as opposed to the earlier PKI based approaches. Their approach however is not suited to delay-sensitive vehicle-to-vehicle communication as vehicles have to contact a base station to decrypt/verify information given by another vehicle.

Golle et al. [68] discuss a different aspect; detecting the presence of malicious data provided by other vehicles. They use a reputation based system to detect erroneous data and propose a method of correcting these errors.

Chapter 6

Concluding Remarks

The future ain't what it used to be.

One of the notable challenges, looming on the horizon, that threatens the successful deployment of sensor networks is privacy. Sensor networks will be deployed to monitor and track valuable assets. In many scenarios, an adversary may be able to backtrace message routing paths to the event source, which can be a serious privacy breach for many monitoring and remote-sensing application scenarios. In this thesis, we have studied the ability of different routing protocols to obfuscate the location of a source sensor. We examined several variations of flooding-based and single-path routing techniques, and found that none of these protocols are capable of providing source location privacy. To achieve improved location privacy, we proposed a new family of routing techniques, called phantom routing, for both the flooding and single-path classes that enhance privacy protection. Phantom routing techniques are desirable since they only marginally increase communication overhead, while achieving significant privacy amplification. We also showed results proving the efficacy of phantom routing protocols in networks of varying characteristics ranging from density, to mobility as well as in the presence of powerful and distributed adversaries.

Just as with location privacy, preventing an adversary from learning the time at which a sensor reading was measured cannot be accomplished by merely using cryptographic security mechanisms. In this thesis, we have proposed a technique complimentary to conventional security techniques that involves the introduction of additional delay in the store-and-forward buffers within the sensor network. We formulated the objective of temporal privacy using an information-theoretic framework, and then examined the effect that additional delay has on buffer occupancy within the sensor network. Temporal privacy and buffer utilization were shown to be objectives that conflict, and to effectively manage the tradeoffs between these design objectives, we proposed an adaptive buffering algorithm, RCAD (Rate-Controlled Adaptive Delaying) that preemptively releases packets under buffer saturation. We then evaluated RCAD using an event-driven simulation study for a large-scale sensor network. We observed that, when compared with a baseline network consisting of unlimited buffers, RCAD was able to provide enhanced temporal privacy (the adversary had higher error in estimating packet creation times), while reducing the end-to-end delivery latency. Further, by adopting variable delays among the nodes along a routing path, we can reduce the number of buffer preemptions in RCAD without noticeably affecting privacy and network performance. Finally, we also devised an improved adversary model that can better estimate the delays produced by RCAD when compared to a naive adversary. In spite of the improved adversary model, RCAD is still able to protect the temporal privacy of sensor flows.

Traffic in wireless sensor networks follows distinctive patterns depending on what events of interest occur. An adversary, armed with knowledge about the type of applications and events in a sensor network, passively observing traffic in the network may be able to infer information about the type of event that occurred simply by observing the packet size and/or the number of packets in succession. This violation of privacy will occur despite always encrypting the event data. In this thesis, we presented a formal definition of the traffic privacy problem, an information theoretic formulation and a quantifiable metric to measure it. We also presented parsimonious bit padding strategies geared towards improving traffic privacy while keeping a check on the overhead in terms of extra bits padded onto every packet. We then described a greedy algorithm that works to provide the highest possible privacy amplification given a bit budget to expend on artificial packet padding.

To the best of our knowledge the work presented in this thesis was one of the earliest works in the area of privacy in sensor networks. We defined the notion of contextual privacy issues in sensor networks and provided ways to mathematically formulate privacy in terms of measurable metrics and showed ways to enhance it. Subsequently, several other researchers have either expanded our works or have contributed new ideas inspired them.

Finally, we designed an identity-based privacy and security framework for vehicular ad hoc networks. We have shown how this framework satisfies the security and privacy requirements of such networks and has an edge over traditional PKI based or symmetric key based systems in terms of scalability, communication overhead and storage requirements while making it easy to achieve confidentiality, pseudonymity and non-repudiation. Future work in this direction includes integrating this framework as a part of various layers of the vehicular network stack including routing algorithms and service and application protocols to address specific challenges in improving the practicability and utility of this framework as well as that of the network. Other directions of research would involve building new applications that use the quantifiable level of trust and privacy that this framework provides.

References

[1] Business week 2002. http://www.businessweek.com.

[2] Vanu. http://www.vanu.com.

[3] A. Perrig, R. Szewczyk, D. Tygar, V. Wen, and D. Culler. SPINS: security protocols for sensor networks. *Wireless Networks*, 8(5):521–534, 2002.

[4] L. Eschenaur and V. Gligor. A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47, 2002.

[5] M. Bohge and W. Trappe. An authentication framework for hierarchical ad hoc sensor networks. In *Proc. of the 2003 ACM Workshop on Wireless Security*, pages 79–87, 2003.

[6] M.Reed, P. Syverson, and D. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16:482–494, May 1998.

[7] P. Syverson, M. Reed, and D. Goldschlag. Onion routing access configurations. In *DISCEX 2000: Proceedings of the DARPA Information Survivability Conference and Exposition*, pages 34–40, January 2000.

[8] M. Reiter and A. Rubin. Crowds: anonymity for web transactions. *ACM Transactions on Information and System Security*, 1:66–92, 1998.

[9] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24:84–88, 1981.

[10] Mixmaster remailer. http://mixmaster.sourceforge.net/.

[11] Marco Gruteser and Dirk Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2003.

[12] WWWF - the conservation organization. http://www.panda.org/.

[13] Wade Trappe and Lawrence Washington. *Introduction to Cryptography with Coding Theory*. Prentice Hall, 2002.

[14] A. Cerpa and D. Estrin. ASCENT: Adaptive Self-Configuring Sensor Networks Topologies. In *Proceedings of IEEE INFOCOM'02*, June 2002.

[15] W. Ye, J. Heidemann, and D. Estrin. An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proceedings of IEEE INFOCOM'02*, June 2002.

[16] C. L. Barrett, S. J. Eidenbenz, L. Kroc, M. Marathe, and J. P. Smit. Parametric probabilistic sensor network routing. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, 2003.

[17] Z. Cheng and W. Heinzelman. Flooding Strategy for Target Discovery in Wireless Networks. In *proceedings of the Sixth ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM 2003)*, 2003.

[18] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM)*, August 2000.

[19] H. Lim and C. Kim. Flooding in Wireless Ad-hoc Networks. In *IEEE computer communications*, 2000.

[20] D. Braginsky and D. Estrin. Rumor routing algorthim for sensor networks . In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, 2002.

[21] P. Th. Eugster, R. Guerraoui, S. B. Handurukande, P. Kouznetsov, and A.-M. Kermarrec. Lightweight probabilistic broadcast. *ACM Transactions on Computer Systems (TOCS)*, 21(4):341 – 374, November 2003.

[22] B. Karp and H. T. Kung. GPSR: greedy perimeter stateless routing for wireless networks. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM)*, August 2000.

[23] D. Niculescu and B. Nath. Trajectory Based Forwarding and its Applications. In *Proceedings of the Ninth Annual ACM/IEEE International Conference on Mobile Computing and Networks (MobiCOM)*, pages 260–272, September 2003.

[24] Roy Yates and David Goodman. *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers, 2nd Edition*. Wiley Publications, 2005.

[25] S. Duri, M. Gruteser, X. Liu, P. Moskowitz, R. Perez, M. Singh, and J. Tang. Context and Location: Framework for security and privacy in automotive telematics. In *Proceedings of the 2nd international workshop on Mobile commerce*, 2002.

[26] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 439–450. ACM Press, May 2000.

[27] Chong K. Liew, Uinam J. Choi, and Chung J. Liew. A data distortion by probability distribution. *ACM Transactions on Database Systems*, 10(3):395–411, 1985.

[28] Naftaly Minsky. Intentional resolution of privacy protection in database systems. *Communications of the ACM*, 19(3):148–159, 1976.

[29] Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2001.

[30] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: attacks and countermeasures. In *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, 2003.

[31] S. Zhu, S. Setia, and S. Jajodia. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of the 10th ACM conference on Computer and communication security*, pages 62–72, October 2003.

[32] H. Chan and A. Perrig. Security and Privacy in Sensor Networks . *IEEE Computer*, 36(10):103–105, October 2003.

[33] Vincent Tseng and Kawuu Lin. Mining temporal moving patterns in object tracking sensor networks. In *International Workshop on Ubiquitous Data Management, 2005.*, 2005.

[34] Pandurang Kamat, Yanyong Zhang, Wade Trappe, and Celal Ozturk. Enhancing source-location privacy in sensor network routing. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, 2005.

[35] Celal Ozturk, Yanyong Zhang, and Wade Trappe. Source-location privacy in energy-constrained sensor network routing. In *Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks*, 2004.

[36] J. Deng, R. Han, and S. Mishra. Intrusion tolerance and anti-traffic analysis strategies for wireless sensor networks. In *IEEE International Conference on Dependable Systems and Networks (DSN)*, 2004.

[37] J Deng, R Han, and S. Mishra. Countermeasures against traffic analysis attacks in wireless sensor networks. In *First IEEE/CreateNet Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm)*, 2005.

[38] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 214–226. ACM Press, 2004.

[39] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27, 2003.

[40] T. Cover and J. Thomas. *Elements of Information Theory.* John Wiley and Sons, 1991.

[41] V. Anantharam and S. Verdu. Bits through queues. *IEEE Trans. on Information Theory*, 42:4–18, 1996.

[42] D. Bertsekas and R. Gallager. *Data Networks.* Prentice Hall, 1992.

[43] D. Guo, S. Shamai, and S. Verdu. Mutual information and minimum mean-square error in gaussian channels. *IEEE Trans. on Information Theory*, 51(4):1261–1282, 2005.

[44] Claudia Diaz and Bart Preneel. Taxonomy of mixes and dummy traffic. In *3rd Working Conference on Privacy and Anonymity in Networked and Distributed Systems*, 2004.

[45] Dogan Kesdogan, Jan Egner, and Roland Buschkes. Stop-and-go-mixes providing probabilistic anonymity in an open system. In *Proceedings of the Second International Workshop on Information Hiding*, pages 83–98, London, UK, 1998. Springer-Verlag.

[46] G. Danezis. The traffic analysis of continuous-time mixes. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies (PET 2004)*, May 2004.

[47] Baik Hoh and Marco Gruteser. Protecting location privacy through path confusion. In *Proceedings of Second International Conference on Security and Privacy in Communication Networks(SecureComm)*, 2006.

[48] Pandurang Kamat, Wenyuan Xu, Yanyong Zhang, and Wade Trappe. Temporal privacy in wireless sensor networks. In *Proceedings of the 27th IEEE International Conference on Distributed Computing Systems (ICDCS'07)*, 2007.

[49] Jong Youl Choi, Markus Jakobsson, and Susanne Wetzel. Balancing auditability and privacy in vehicular networks. In *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*, pages 79–87, New York, NY, USA, 2005. ACM Press.

[50] Jean-Pierre Hubaux, Srdjan Capkun, and Jun Luo. The security and privacy of smart vehicles. *IEEE Security and Privacy*, 2(3):49–55, 2004.

[51] Maxim Raya and Jean-Pierre Hubaux. The security of vehicular ad hoc networks. In *SASN '05: Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks*, pages 11–21, New York, NY, USA, 2005. ACM Press.

[52] Wenyuan Xu, Wade Trappe, Yanyong Zhang, and Timothy Wood. The feasibility of launching and detecting jamming attacks in wireless networks. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 46–57, 2005.

[53] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Proceedings of CRYPTO 84 on Advances in cryptology*, pages 47–53, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[54] Dan Boneh and Matt Franklin. Identity-based encryption from the Weil pairing. *Lecture Notes in Computer Science*, 2139, 2001.

[55] Yuliang Zheng. Digital signcryption or how to achieve cost (signature & encryption) $<<$ cost(signature) $+$ cost(encryption). *Lecture Notes in Computer Science*, 1294:165–179, 1997.

[56] L. Chen and J. Malone-Lee. Improved identity-based signcryption, 2004.

[57] Multiprecision integer and rational arithmetic c/c++ library (miracl). Shamus Software Ltd.

[58] Dedicated short range communication standard (dsrc). http://grouper.ieee.org/groups/scc32/dsrc/namerica/index.html.

[59] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *Wireless Communications and Networking Conference*, 2005.

[60] Manpreet Singh, Maximilian Ott, Ivan Seskar, and Pandurang Kamat. Orbit measurements framework and library (oml): Motivations, design,implementation, and features. In *Proceedings of IEEE Tridentcom, Trento, Italy*, 2005.

[61] P. Chatzimisios, V. Vitsas, and A. C. Boucouvalas. Throughput and delay analysis of ieee 802.11 protocol. In *IEEE 5th International Workshop on Networked Appliances*, 2002.

[62] Marios D. Dikaiakos, Saif Iqbal, Tamer Nadeem, and Liviu Iftode. Vitp: an information transfer protocol for vehicular computing. In *VANET '05: Proceedings of the 2nd ACM international workshop on Vehicular ad hoc networks*, pages 30–39, New York, NY, USA, 2005. ACM Press.

[63] Maxim Raya and Jean-Pierre Hubaux. Security aspects of inter-vehicle communications. In *Swiss Transport Research Conference*, 2005.

[64] Bryan Parno and Adrian Perrig. Challenges in securing vehicular networks. In *Proceedings of Workshop on Hot Topics in Networks (HotNets-IV)*, November 2005.

[65] Magda El Zarki, Sharad Mehrotra, Gene Tsudik, and Nalini Venkatasubramanian. Security issues in a future vehicular network. In *European Wireless*, 2002.

[66] Florian Doetzer. Privacy issues in vehicular ad hoc networks. In *Workshop on Privacy Enhancing Technologies*, 2005.

[67] Jeremy Blum and Azim Eskandarian. The threat of intelligent collisions. *IT Professional*, 2004.

[68] Philippe Golle, Dan Greene, and Jessica Staddon. Detecting and correcting malicious data in vanets. In *VANET '04: Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, pages 29–37, New York, NY, USA, 2004. ACM Press.

Curriculum Vita

Pandurang Kamat

Education

1993-1997 B.E. in Computer Engineering; Goa Engineering College, Goa University, India.

1998-1999 M.S. in Computer Science; New Jersey Institute of Technology, NJ, USA.

2003-2007 Ph.D. in Computer Science; Rutgers University, NJ, USA.

Industry

1999-2000 Principal Software Engineer, FoxyTrader Inc., NJ, USA

2000-2002 Member of Technical Staff, Bell Labs - Lucent Technologies, NJ, USA

2004 Researcher, Hewlett Packard Labs, NJ, USA

2005 Visiting Researcher, National Institute of Information and Communication Technology, Japan.

2007- Senior Research Engineer, Ask.com, NJ, USA.

Publications

2007 P. Kamat, W. Xu, Y. Zhang and W. Trappe, *Temporal Privacy in wireless sensor networks*. In proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS) 2007. Acceptance rate 13

A. Baliga, P. Kamat, and L. Iftode, *Lurking in the Shadows: Identifying systemic threats to kernel data*, (short paper). In proceedings of IEEE Symposium on Security and Privacy, Oakland, 2007.

P. Kamat, A. Baliga and W. Trappe, *Secure, pseudonymous and auditable communication in vehicular networks*. Under Submission.

P. Kamat, Y. Zhang, and W. Trappe, *Source location privacy in wireless sensor networks*. Under Submission.

R. Miller, P. Kamat, W. Xu and W. Trappe, *Service discovery and device identification in cognitive radio networks*. In proceedings of IEEE Workshop on Networking Technologies for Software Defined Radio Networks, to be held with IEEE SECON 2007.

S. Paul, S. Ganu, P. Kamat and E. B. Royer, *Requirements for wireless GENI experiment control and management*. Global Environment for Network Innovations (GENI) design document GDD-07-43.

2006 P. Kamat, A. Baliga and W. Trappe, *An identity-based security framework for VANETs*, (short paper). In proceedings of ACM International Workshop on Vehicular Ad Hoc Networks (VANET) held with MOBICOM 2006.

W. Xu, P. Kamat and W. Trappe, *TRIESTE: A trusted radio framework for enforcing spectrum etiquettes*. In proceedings of IEEE Workshop on Networking Technologies for Software Defined Radio Networks, held with IEEE SECON 2006.

S. Haber and P. Kamat, *A content integrity service for long-term digital archives*. In proceedings of IS&T Archiving Conference (Archiving 2006), Ottawa, Canada. May 2006.

2005 P. Kamat, Y. Zhang, W. Trappe and C. Ozturk, *Enhancing source-location privacy in sensor network routing*. In proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS) 2005. Acceptance rate 13

M. Singh, M. Ott, I. Seskar and P. Kamat, *ORBIT measurements framework and library (OML): motivations, design, implementation and features*. In proceedings of IEEE Tridentcom 2005.

1996 P. Kamat and B. Gonsalves, *Enterprise Resource Planning*, In proceedings of National conference on Information Technology for the 21st century, Surathkal, India, 1996. (Best paper award)

Reviewer

IEEE Journal on Selected Areas in Communications (JSAC).

EURASIP Journal on Information Security.

IEEE Wireless Communications Magazine Special Issue on Security in Wireless Mobile Ad Hoc and Sensor Networks, 2007.

IEEE Conference on Computer Communications (Infocom), 2007 and 2008.

IEEE Sarnoff symposium, 2007.

IEEE Wireless Communications & Networking Conference (WCNC), 2005, 2006 and 2007.

International Conference on Information and Communications Security (ICICS), 2006.

IEEE Communication Letters.

SPEUCS 2007: Workshop on the Security and Privacy of Emerging Ubiquitous Communication Systems.