12-1-2012

Energy conserving privacy enhancing algorithms for resource-constrained devices

Michael Groat

Michael Matthew Groat

Candidate

Department of Computer Science

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Stephanie Forrest, Ph.D. , Chairperson

Wenbo He, Ph.D.

Carlos Fernando Esponda Darlington, Ph.D.

Terran Lane, Ph.D.

Jared Saia, Ph.D.

Energy Conserving Privacy Enhancing Algorithms for Resource-Constrained Devices

by

Michael Matthew Groat

B.S., Computer Science, California State University, East Bay, 1997
M.S., Computer Science, California State University, East Bay, 2005

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy
Computer Science

The University of New Mexico

Albuquerque, New Mexico

December, 2012

Dedication

To the disadvantaged among us,
and the people that support them.

Acknowledgments

I would first like to thank my adviser, Stephanie Forrest, for her guidance, counseling, and opportunity to work as her research assistant. I thank my co-adviser, Wenbo He, for her inspirational source of technical contributions, and, I thank the rest of my committee for providing helpful and constructive advice: Fernando Esponda, Terran Lane, and Jared Saia.

Much recognition goes to the members of the Adaptive Computation Laboratory for their help, discussions, and support: Elena Ackley, George Bezerra, James Horey, Josh Karlin, Drew Levin, ThanhVu Nguyen, Terri Oda, Barbara Shirtcliff, Eric Shulte, George Stelle, and Eric Trias. I would like to thank especially Benjamin Edwards for his help, ideas, and assistance with the PerCom conference, and Ricardo Villalon for his assistance with walking me through NesC, TinyOS, and TOSSIM.

I acknowledge and thank the various anonymous reviewers who helped shape the content of the published papers that assisted in forming this dissertation. I also acknowledge financial support from Motorola Inc., Reaching the Pinnacle, the National Science Foundation, the Defense Advanced Research Projects Agency, and Eli Lilly and Company.

This dissertation could not be completed without the support and encouragement from my family, and, thank you Carol for your patience, wisdom and love. Most of all, I thank God for inspiring and guiding me through graduate school.

Energy Conserving Privacy Enhancing Algorithms for Resource-Constrained Devices

by

Michael Matthew Groat

B.S., Computer Science, California State University, East Bay, 1997
M.S., Computer Science, California State University, East Bay, 2005
Ph.D., Computer Science, University of New Mexico, 2012

Abstract

Resource-constrained devices such as wireless sensor networks, body area networks, or smart phones collect confidential and sensitive information about their users. Traditional solutions to protect these data, such as encryption, consume a significant amount of resources to be viable. In this dissertation, I present two energy efficient information collection protocols based on the notion that by relaxing the definition of privacy, such as using indistinguishability, energy use can be reduced. The first protocol, *multi-dimensional negative surveys (MDNSs)*, protects multivariate categorical data by perturbing sensed values to something other than what was actually sensed, and transmits the perturbed values to a central information collection server, providing privacy protection for information such as location. The second protocol, *k-indistinguishable privacy-preserving data aggregation (KIPDA)*, protects the privacy of data that are aggregated in wireless sensor networks. It is specialized for the maximum and minimum aggregation functions and is one of the first techniques to provide protection from other adversarial nodes in the network. Sensitive data are obfuscated by hiding them among a set of camouflage values. Because the sensitive data are not encrypted, they can be aggregated easily and efficiently with minimal in-network processing delay. While radio usage is expensive, I show through analysis, simulations, and implementations that broadcasting a modest amount of camouflage data is more energy efficient when encryption is eliminated. Simulations and implementations on physical devices illustrate how both approaches can protect the privacy of a participant's data, while reducing energy use and allowing useful aggregate information to be collected.

Contents

List of Figures

List of Tables

Glossary

α_i Number of categories in dimension i of a negative survey. If no subscript is used, then only a single dimension exists.

A The reconstructed distribution in a negative survey.

aggregation tree The path through a wireless sensor network (WSN) that data take to reach the base station. This path typically resembles the minimum spanning tree of a WSN, with the base station at the root.

base station A central information collection server that receives data in a WSN. A WSN can have more than one base station.

CDA Concealed data aggregation. Data aggregation where data are kept confidential and secure from intermediate nodes and eavesdroppers.

d_i Sensitive value of data at node i. It is hidden in among $|V^i|-1$ other values in node i's message vector, V^i.

D Number of dimensions in a multi-dimensional negative survey (MDNS).

DA

Dimensional adjustment. A technique that increases the number of dimensions and reduces the number of categories in a single dimension. This increases utility and reduces privacy non-linearly, with a higher increase in utility than the corresponding decrease in privacy.

data aggregation

The combining, merging, processing, or filtering of data while in transit to the base station. This reduces packet size and number of packets, thus conserving energy.

GSS

Global secret set. In KIPDA, the set stored at the base station that contains possible locations for the final network aggregated value.

honest but curious

A threat model that assumes an entity will follow the network protocols but will use the protocols to collect information mischievously.

I

Index set of $V^i,\ \forall i,\ 1 \leq i \leq N$, where N is the number of sensors in a wireless sensor network (WSN). $I = \{1, 2, ..., n\}$, where n is the number of messages in a message vector.

KIPDA

k-Indistinguishable privacy-preserving data aggregation. A CDA scheme in WSNs where data are camouflaged among decoy values.

Kronecker technique

A technique that converts a multi-dimensional negative survey (MDNS) to a single-dimensional negative survey, with the same accuracy.

LPL

Low power listening. A technique used by sensors to conserve energy by putting the radio to sleep. The radio wakes randomly to listen for transmissions.

M_i Notation for the perturbation matrix of dimension i. If no subscript is used, then only a single dimension exists.

MDNS Multi-dimensional negative survey. An information collection scheme that perturbs multi-dimensional sensor data to obscure the data, but allows aggregate statistics to be reconstructed from the perturbed values.

message vector Vector of the camouflaged and sensitive values sent to the next hop in the aggregation tree. It is indexed by I.

n Number of values in a message vector, $n = |V^i|, \ \forall i, \ 1 \leq i \leq N$.

N Number of participants in a negative survey or number of sensors in a wireless sensor network (WSN).

negative survey A survey in which respondents (human or device) report a value other than the correct value. For example, if a respondent drove a Chevrolet and was asked what make of car she drove, she might report a Ford, but would not report a Chevrolet.

NSPM Negative survey perturbation matrix. An α by α probability matrix that maps category i to category j in a negative survey. It contains 0's down the diagonal and $\frac{1}{\alpha - 1}$ everywhere else.

P^i Index of the sensitive value in V^i for node i. $P^i \subset R^i$.

perturbation matrix The matrix in random response techniques (RRTs) and negative surveys that gives the probabilities of perturbing a category to another category. In MDNS each dimension has a NSPM.

R^i Indices of the restricted camouflage values in V^i for node i with the union of the index of the sensitive value, P^i.

RDP Random data perturbation. A technique in privacy-preserving data mining in which the original data is perturbed by adding noise drawn from a known distribution. The perturbed data is then reconstructed using an iterative algorithm based on Bayes Theorem.

restricted values Camouflaged values in the message vector that are greater than the sensitive value for minimum aggregation (and less for maximum aggregation).

RRT Random response technique. A method in privacy-preserving data mining that perturbs categorical information. Originally developed to hide answers to yes or no questions, it has been extended to multiple categories.

sensor A small resource-constrained device typically equipped with a radio of limited bandwidth, a simple central processing unit, a limited amount of memory (either volatile and/or non-volatile), a limited amount of energy, and one or more environmental sensors. Used synonymously with node or mote.

U^i Indices of the unrestricted camouflaged values in V^i for node i. $U^i = \overline{R^i} = V^i - R^i$, where "$-$" denotes set difference.

unrestricted values Camouflage values in a message vector that are either more or less than the sensitive value in the message vector.

v^i_ℓ Values of V^i for node i where $\ell = 1, 2, ..., n$.

V^i Notation for the message vector of node i. $V^i = \{v_1^i, v_2^i, ..., v_n^i\}$.

V^Ω The last message vector received by, or processed at, the base station.

WSN Wireless sensor network. A network of sensors that monitors its environment and communicates by radio transmissions.

X The original or underlying distribution in a negative survey.

Y The perturbed, disguised, or negated distribution in a negative survey.

Chapter 1

Introduction [1]

Applications of sensor networks have shifted from monitoring non-sensitive data about volcanoes [148] and forests [14] to collecting private and confidential information about people's health, habits, and behaviors [27]. Because sensors now interact closely with people, it is vital to protect and secure the data they collect. However, current forms of protection are expensive and consume resources such as energy stored in batteries, processing time, and memory [97, 124, 144]. New forms of privacy protection are required to alleviate this strain and protect sensitive data on resource-constrained devices [135]. I propose that algorithms can trade-off strict notions of privacy for data indistinguishability, reducing energy consumption as a consequence.

[1] Some material from this dissertation was previously published in "Enhancing Participatory Sensing Applications with Multidimensional Data" which appeared in the *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications* ©IEEE 2012, and "KIPDA: k-Indistinguishable Privacy-preserving Data Aggregation in Wireless Sensor Networks" which appeared in the *Proceedings of the 30th IEEE International Conference on Computer Communications* ©IEEE 2011. Reprinted with permission.

Novel methods of energy efficient privacy protection are important for several reasons. First, privacy protects individuals from a variety of negative consequences they may suffer when data are leaked about them, for example, thefts of bank accounts or social security numbers. The second reason is economical — relieving the load on sensor resources prolongs the lifetime of these devices and the intervals between needed maintenance [12]. Third, the growth of ubiquitous sensing will affect nearly everyone. Sensors will exist in abundance in many commonplace locations such as milk cartons [20], light bulbs [66], clothes [119], street lights [93], buses [72], and bathrooms [116]. These devices may inadvertently capture information from unwilling participants. Finally, while a small sensor may not consume a large amount of energy (especially when compared to a personal computer), their pervasiveness will eventually affect energy budgets.

Creating new forms of privacy protection is not easy because of several challenges. Eavesdroppers could intercept sensor radio communications. The devices themselves can be physically captured by an adversary and have their memory contents examined, or they can be altered and re-released in the environment to masquerade as benign sensors, mischievously collecting information or disrupting protocols. In addition, there are several impediments to creating new privacy protection schemes:

- **Encryption is expensive:** Because of the limited resources of sensors, mainstream data privacy solutions such as asymmetric (public/private key) encryption are too memory and power intensive [97, 124, 144, 145]. Although some research has demonstrated that certain types of symmetric encryption are possible [96, 125, 146], there remains a significant computational cost and other issues such as key distribution and management [57]. For example, TinySec, a common implementation of security protocols on sensors (that uses symmetric encryption) consumes 10% more energy than a comparable implementation without the security protocols [96]. This is a conservative estimate given by the creators of TinySec [96]. Although not huge, it can reduce a year of battery life by at least a month.

- **Encryption must ultimately trust the final recipient:** All encryption schemes eventually trust some person or entity with the decrypted information. In some applications, this may not be desirable [85], e.g., consider a vehicle's speedometer and GPS sensor that report a speed of 80 miles per hour on an open freeway. Encryption protects a message in transit to its destination. However, once the message reaches its destination, its information can still be compromised (even if the final recipient is trusted) through theft, negligence, or search warrants. An important research challenge is protecting information in this broader sense while allowing certain operations and queries over the data.

- **Timing issues:** Time delay is a common problem with encryption algorithms on sensors [16]. The time to send one byte over a radio is informally known as the *byte time*. If a node cannot encrypt the next byte in this amount of time, the radio will have to wait, causing delay. If sensors are using *low power listening (LPL)* [94], delay in the network can increase the amount of energy consumed [42]. LPL puts a sensor's radio in a low power state to conserve energy. Additionally, the extra bytes from encryption protocol overheads (such as ciphertext block sizes larger than plaintext sizes) increase packet length, adding to time delay and energy consumption.

- **Networks:** The topology of sensor networks create an additional complication because sensors typically route information over a wireless radio through each other to reach a *base station*, or central information collection server. For example, a node may transmit data to a base station that must be concealed from the other nodes in transit. Yet, these other nodes may be configured to perform certain tasks such as an aggregation of the sum or an average, which are more easily accomplished with plaintext messages. This problem is known as *concealed data aggregation (CDA)* [24, 25, 26, 29, 65, 73, 83, 128, 151] and is discussed in more detail in Chapter 2, Section 2.

These four challenges: encryption costs, the need for universal privacy, time delays, and the network topology of sensors are the focus of this dissertation.

Literature in the field that addresses these challenges is slowly moving away from purely cryptographic solutions [24, 26]. This supports my argument that encryption is not a panacea for every data privacy situation. Additionally, it suggests that a trade-off may exist between the privacy level and the computation and communication resources.

This dissertation presents two new information collection protocols that protect privacy and reduce energy usage. The first protocol, known as *multi-dimensional negative surveys (MDNSs)*, perturbs multivariate categorical sensed information and reports the perturbed values to a centralized collector. The perturbation technique has certain properties that allow statistics about the original sensed data to be reconstructed from the consolidated perturbed data. Improvements to the protocol are described that reduce the number of samples needed to achieve accurate reconstruction. Two metrics are proposed for measuring (1) the amount of privacy a node will have against an adversary, and (2) the utility of the reconstructed data.

The second protocol, known as *k-indistinguishable privacy-preserving data aggregation (KIPDA)*, camouflages sensitive data among decoy values. While some energy is spent transmitting decoy values over the radio, analysis and simulations show net energy savings because encryption is avoided. KIPDA is a concealed data aggregation scheme because it allows sensors to perform aggregation functions without knowing the sensitive values. The protocol is among the first to protect sensitive data in MAX/MIN data aggregation from other in-network nodes, providing protection from node collusion and node capture attacks.

The MDNS protocols were simulated in MATLAB, while the KIPDA protocols were simulated in an energy aware TinyOS Simulator, PowerTOSSIM-Z. Both protocols were implemented on real physical devices: Moteiv T-Mote Invent sensors for KIPDA, and Android smart phones for MDNSs. These simulations and implementations demonstrate the feasibility of the protocols, suggest possible applications, and compare the resource savings to privacy levels.

Several aspects of the algorithms are counter-intuitive, such as spending more energy to communicate extra bytes over the radio, or reporting false information. The ideas were inspired by natural processes: the natural human immune system's method of negative selection (MDNS), and an animal's use of camouflage to protect themselves from predators (KIPDA). In the human immune system, the thymus educates T-cells to attack foreign cells or cells infected with foreign viruses. In the MDNS protocol, data are perturbed to something other than self. Camouflage allows an animal to blend into its environment to hide from predators. KIPDA inserts sensitive data into a message vector to hide the value from adversaries. It is also similar to immunocamouflage [32] which coats red blood cells with a polymer to allow blood transfusions with non-compatible patients. Similarly, KIPDA "coats" the sensitive data with extra information so that an adversary cannot attack or determine the sensitive value. My solutions are appropriate for situations where the maximum protection provided by encryption is not required, and where resource constraints exist.

The main contributions of this dissertation include:

1. An information collection protocol, MDNSs, that can disguise efficiently multivariate categorical data and allow statistics to be reconstructed from the corpus of disguised data.

2. A technique called *dimensional adjustment* that improves MDNSs by reducing

the number of samples required for accurate reconstruction, allowing practical applications.

3. The first comparison of MDNSs to a popular perturbation technique, random data perturbation (RDP), used in privacy-preserving data mining that operates on continuous data.

4. A second information collection protocol, KIPDA, that camouflages sensitive data among decoy values and is one of the first secure comparison techniques that protects information from the nodes that perform aggregation.

5. Simulations in MATLAB and TOSSIM that illustrate potential applications and quantify the levels of privacy attained against resource effectiveness.

6. Implementations on physical devices that demonstrate the feasibility of the two information collection protocols.

Roadmap: The remainder of this dissertation is outlined as follows. Chapter 2 gives background material on wireless sensor networks, privacy-preserving data aggregation, and negative surveys, and defines the assumptions used throughout the rest of this work. Chapters 3 and 4 introduce the MDNS and KIPDA protocols respectively. Chapter 5 illustrates the two protocols with several MATLAB and TOSSIM simulations, and presents real world implementations on Moteiv's T-Mote Invent sensors and Android smart phones. Chapter 6 provides a discussion and Chapter 7 compares the two protocols with related work. The final chapter suggests future work and gives the conclusion.

Chapter 2

Background and Preliminary Work

This chapter states the assumptions that are used throughout the rest of the dissertation and provides an overview of wireless sensor networks, concealed data aggregation, and negative surveys. Preliminary work that studies negative surveys applied to wireless sensor network is also presented.

2.1 Wireless Sensor Networks

A *sensor* is a small device typically equipped with a radio transmitter, a small micro-controller, one or more environmental sensors, some type of memory, and a power source such as a battery or solar cell. The terms sensor, node, and mote are used interchangeably. Sensors can vary in size from that of a loaf of bread to that of a grain of rice or even dust, and because of their size and cost constraints, are typically resource limited, including power, bandwidth, memory, and computational ability [135]. When used together, they create a wireless sensor network.

A *wireless sensor network (WSN)* is a spatially distributed network of resource-constrained sensors. Each node is autonomous, yet the collection of nodes coop-

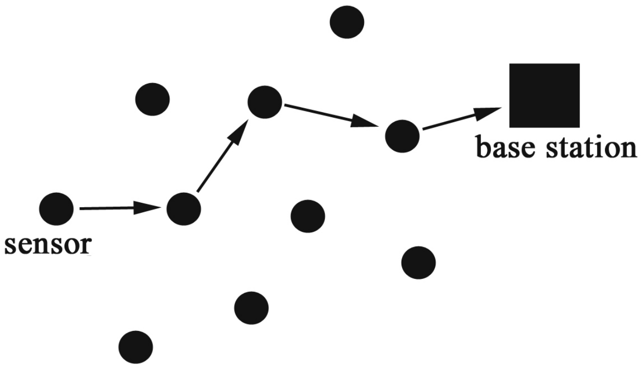


Figure 2.1: Example of a wireless sensor network. Because of the limited radio range of the sensor nodes, they route information through each other to reach a base station.

eratively monitor certain physical or environmental conditions such as temperature, pressure, pollutants, or sound. Sensors communicate directly with each other through radio transmissions, but because of their limited radio range, nodes must route packets through their neighbors to reach one or more base stations, which are not typically as resource-constrained. Figure 2.1 illustrates such a network. Although originally developed for military battlefield use [101], they are now widely used by civilians in many areas such as health care [36, 82, 98], geological surveys [148], and industrial [4], habitat [14, 27], structural [104], or traffic [34, 85] monitoring. WSNs are modeled in this dissertation as a connected graph $G(\mathcal{V}, \mathcal{E})$, where sensor nodes are represented as vertices $\mathcal{V}$ and wireless links as edges $\mathcal{E}$. The number of sensor nodes is defined as $N = |\mathcal{V}|$.

Real world WSNs can deviate from this model in various ways. For example, in the cell phone radiation detection simulation presented in Chapter 5, each node communicates directly to the base station through cell phone towers. However, these devices are still resource-constrained. Chapter 4 returns to a more traditional WSN architecture and examines the problem of concealed data aggregation, discussed in the next section.

2.2 Concealed Data Aggregation

WSNs designers employ a common technique called data aggregation to conserve energy [1, 33, 44, 91, 92, 104, 137, 141]. Ideally, each node should report its entire data set to the base station. However, this large amount of information can drain the network of energy. If the base station does not need every measurement by every node, WSNs can perform in-network processing on the data along its path to the base station. Nodes can combine, change, filter, or process measurements to limit the amount of data transmitted over the radio. For example, if a user is only interested in the sum of the sensed values over a certain time period, nodes can sum the information they receive and pass that information to the next node closest to the base station. To accomplish this, routes in an aggregation scheme typically follow a tree structure [109], such as the minimum spanning tree. Figure 2.2 illustrates such a route and aggregation process. The sizes of the arrows are proportional to the amount of information transmitted. On the left side, each node sends its sensed value to the base station. On the right side, data are aggregated and the amount of traffic is reduced. The data aggregation function is defined in this dissertation as $y(t) \triangleq f(d_1(t), d_2(t), \cdots, d_N(t))$, where $d_i(t)$ is the individual sensor reading at time t for node i.

Data aggregation is important because it saves energy by reducing the number of packets and packet lengths. Every bit transmitted over a radio uses an equivalent amount of energy to that for 800 to 2,000 clock cycles of execution on a microcontroller, depending on the architecture [145] and the distance to transmit. Consequently, reducing the number of bits that are transmitted is analogous to reducing

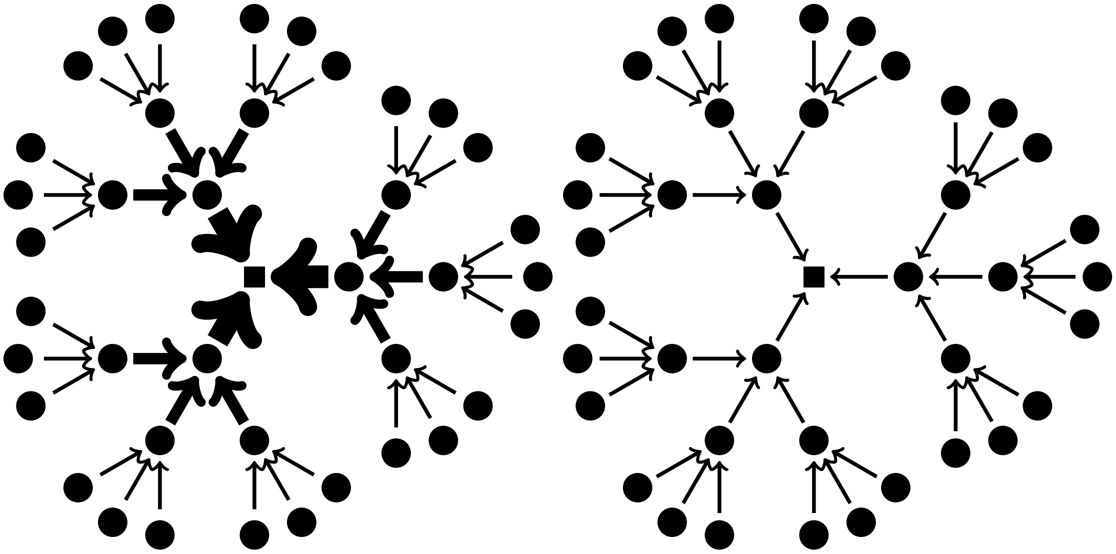


Figure 2.2: Example of a WSN with and without data aggregation. Width of the arrows is proportional to the amount of data transmitted. Routes follow a tree structure in the network. (Left) A WSN in which values are not aggregated. All data values are reported to the base station. (Right) Same network with data aggregation. Nodes combine information from their children to reduce the amount of information transmitted.

the energy consumed.

Data aggregation can be trivially implemented in WSNs. However, it is more challenging when privacy and security are a concern, as information can potentially be disclosed to either outside observers, neighboring nodes in the network, or intermediate nodes performing the aggregation. *Concealed data aggregation (CDA)*, also known as privacy-preserving data aggregation, aggregates data while keeping it confidential and protected [77, 82]. This is not trivial because of the following challenges:

- **Intermediate node ignorance:** Intermediate nodes need to aggregate data without actually knowing the values.

- **Base station ignorance:** Sometimes it may be desirable for the base station to collect information and obtain statistics from sensor nodes without knowing any individual node's information.

- **Non-linear aggregation functions:** Non-liner functions such as MAX and MIN are difficult to securely aggregate [131] because they do not work well with traditional forms of homomorphic encryption which rely on the linear characteristics of polynomials.

- **Energy conservation:** CDA requires conservation of energy for each sensor and the network as a whole to prolong the devices' lifetimes.

Hop-by-hop aggregation [151] is a traditional approach that addresses these four challenges. A node encrypts its sensed information before sending it to the next hop (or parent) in the aggregation routing tree, where it is decrypted and aggregated with other information. This aggregate is then encrypted and passed to the next hop (parent's parent). This technique protects data from outside observers, however, plaintext is available at each node after decryption, which increases the risk of data leakage through node capture attacks. Additionally, extra energy is spent and latency is introduced, due to the repeated decryption and encryption process.

End-to-end encryption proposes solutions to these limitations. A set of algorithms known as *privacy homomorphism* have been developed to aggregate encrypted data without decrypting it [26, 73]. For example, if the aggregation functions are summation or multiplication, then the following properties hold:

$$\begin{aligned} x + y &= Decrypt[Encrypt(x) \oplus Encrypt(y)] \\ x * y &= Decrypt[Encrypt(x) \otimes Encrypt(y)], \end{aligned} \tag{2.1}$$

where $\oplus$ and $\otimes$ are special homomorphic addition and multiplication functions. Because data remain encrypted from one end of the network to the other, the problem of data confidentiality from the intermediate nodes typically does not arise. Additionally, energy is saved because the repeated encryption and decryption phases are avoided. For aggregation functions such as addition and multiplication, CDA has been well addressed in WSNs. For example, Girao et al. [73] use the Domingo-Ferrer's privacy homomorphism to aggregate the average and movement detection functions.

However, research on more general nonlinear aggregation functions such as maximum and minimum has been limited. Rivest et al. [131] showed that homomorphic encryption is insecure to ciphertext only attacks if comparison operators are supported. Attempts have been made to address this limitation with homomorphic encryption based on public key technology [40], but these schemes are too expensive for practical use on WSNs. Acharya et al. [3] efficiently tailored a method called Order Preserving Encryption Scheme (OPES) [8] from databases to WSNs. In their scheme, sensor nodes map their plaintext measurements into a set of ciphered values, which preserves the order of the measurements. Hence, aggregators are able to compare the values and aggregate them without decrypting. Sensors manage to hide the plaintext distribution, which secures the algorithm against ciphertext only attacks. However, the scheme cannot prevent in-network neighbors from learning private data if they use the same set of mapping functions. The KIPDA protocol presented in Chapter 4 is designed specifically for secure comparison aggregation by providing robustness against in-network neighbors from learning private data.

2.3 Negative Surveys

This section introduces negative surveys as background material for Chapter 3. First, I present a generalized case from privacy-preserving data mining called *randomized response techniques (RRTs)* [2, 10, 11, 18, 19, 31, 51, 67, 87, 89, 115, 147]. RRTs disguise data by perturbing a categorical value to another value. For example, in a survey of ethnicity, if a participant is Hispanic, the response might be randomly perturbed to a new value, such as Asian. A *perturbation matrix*, denoted M, gives the probabilities of perturbing category i to category j. It is an $\alpha \times \alpha$ square matrix, where each entry $M_{i,j}$ is the probability of responding with category j when category i is detected.

Finding the optimal M that balances privacy and utility has been the subject of earlier research [11, 89]. Warner described the RRT for binary data [147], which can be extended to categorical data [10] using the following perturbation matrix, which gives an initial suggestion for M:

$$M = \begin{pmatrix} p & \frac{1-p}{\alpha-1} & \cdots \\ \frac{1-p}{\alpha-1} & p & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}, \tag{2.2}$$

where p is the probability that a category remains unchanged. Similar schemes such as the Uniform Perturbation (UP) [10], and Framework for High-accuracy Privacy-preserving Mining (FRAPP) [11] matrices perform similarly to the Warner scheme [89].

The original data are estimated from the disguised data using the following equation [59, 63]:

$$\widehat{A} = M^{-1}\widehat{Y}, \tag{2.3}$$

where $\widehat{Y} = (Y_1, \ldots, Y_\alpha)^\tau$ and Y_i is the number of disguised values in the i^{th} category. Since this is an unbiased maximum likelihood estimate, $\widehat{A}$ approaches the original distribution as the population size increases. Equation (2.3) is known as the matrix inversion approach. An iterative approach is given by Agrawal et al. [10] but is not extended to multiple dimensions.

A special case of the Warner scheme, called negative surveys [59, 63, 85], uses a perturbation matrix containing zeros on the diagonal entries and equal values everywhere else where the columns sum to one, i.e., $p = 0$ in Equation (2.2). I will call these matrices *negative survey perturbation matrices (NSPMs)* throughout the rest of this dissertation.

Intuitively, a negative survey [59, 63, 143] is best explained in the context of a traditional survey. Suppose you are asked which of the following cars you drive: Ford, General Motors, Toyota, Honda, or Chevrolet. Assuming you drive one and only one of the previous makes, then you could "truthfully" report the vehicle you regularly drive. However, if this were instead a list of sexually transmitted diseases, you might be more hesitant to answer. A negative survey asks you to "lie," by reporting a vehicle you do *not* drive. In the case of the sexually transmitted diseases, a person might be more willing to answer the negative survey. An example is illustrated in Figure 2.3.

2.4 Preliminary Work: Negative Surveys Applied to Wireless Sensor Networks

As preliminary work and in collaboration with James Horey [85], negative surveys were applied to WSNs. Two protocols were devised: the node and base station protocols. The first maps sensed data to its negative representation. Each node chooses a category it did not sense with uniform probability and returns that "negative" information to the base station. In the *base station protocol*, the base station collects the negative information from each node and reconstructs an estimate of the original data from the collected data. Instead of Equation (2.3), the following simpler

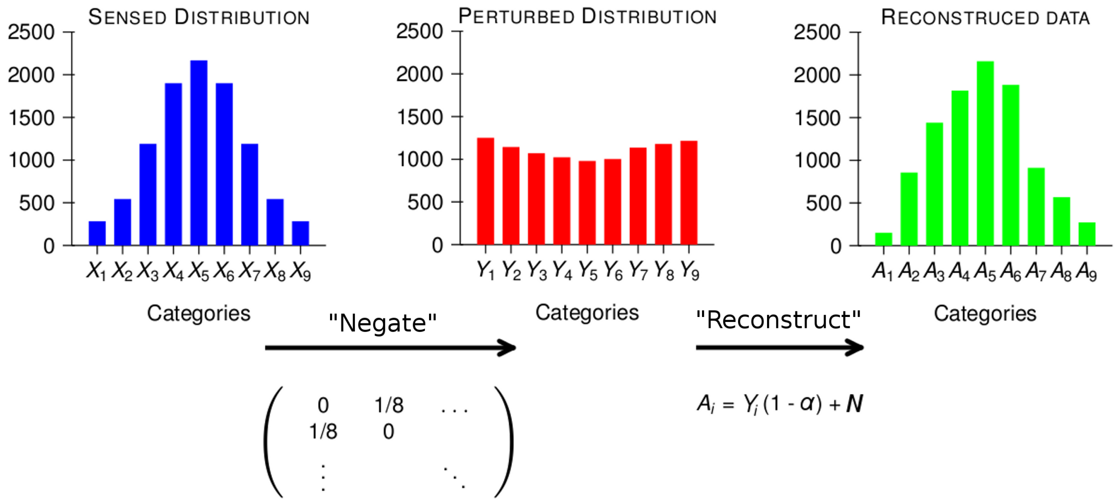


Figure 2.3: Example of a single-dimensional negative survey with 9 categories and 10,000 samples. Each sample from the sensed distribution is perturbed according to the given perturbation matrix. The perturbed data is reconstructed with the given equation where $N = 10,000$ and $\alpha = 9$.

reconstruction equation [59] can be used:

$$A_i = N - (\alpha - 1) \cdot Y_i, \tag{2.4}$$

where A_i is the reconstructed number of values in category i, and Y_i is the reported perturbed number of values in category i, with $1 \leq i \leq \alpha$. N is the total number of sensed values. Equation (2.4) has time complexity $O(\alpha)$, compared to $O(\alpha^2)$ for Equation (2.3) (ignoring matrix inversion), while still remaining an unbiased maximum likelihood estimate.

In Horey et al. [85], the difference between the original and reconstructed distributions, called utility [9, 105], was measured as the relative root mean squared error (rRMSE) given as:

$$rRMSE = \sqrt{\frac{1}{\alpha}\sum_{i=1}^{\alpha}\left(\frac{A_i - X_i}{X_i}\right)^2}, \tag{2.5}$$

where X_i is the original number of sensed values in category i, and A_i and α were previously given. In Chapter 3, utility is measured with the mean square error, and is calculated more accurately with the variance of the negative surveys. Privacy was not calculated in Horey et al. [85], but is calculated in Chapter 3 as the probability of guessing the original data from the disguised values based on the maximum *a posteriori* estimate.

The feasibility of negative surveys was illustrated in Horey et al. [85] with a traffic monitoring simulation where cars reported to a stationary base station a speed at which they were not traveling. Speeds where quantized into categories and the base station reconstructed the perturbed information into histograms of driving speeds. The histograms were accurate enough so that traffic behavior could be correctly classified as either congested, normal, or speedy. It was determined that the method was practical for the current levels of traffic.

The benefits of negative surveys in WSNs with respect to encryption are both efficiency and privacy protection. The time complexity of the node protocol is only a slight constant increase, $O(1)$, from reporting the true sensed value. This is an advantage over even the simplest encryption methods. The base-station (or any other entity) does not have to be trusted, since the information it receives is perturbed. Negative surveys in WSNs also eliminate the need for key distribution and management, which can be problematic [23, 28, 50, 57, 99, 107].

In the next chapter, negative surveys are extended to multivariate categorical data, increasing the range of possible applications, two of which are presented in Chapter 5.

2.5 Assumptions

This section presents the assumptions, threat models and notions of privacy that are used throughout the rest of the dissertation. I distinguish between privacy, or *data confidentiality*, which ensures that data are not discoverable by an adversary in a feasible amount of time, versus security, or *data integrity*, which ensures that data are not sabotaged, changed, or withheld, along the way to its destination. This dissertation addresses data confidentiality, leaving data integrity for future work.

My proposed solutions sometimes provide less privacy than standard cryptography, such as adversaries who have partial knowledge of the secure information, or a datum that is indistinguishable from other camouflage data. In other cases, my notions of privacy are stronger than cryptography, such as the MDNS protocols that protect data once it leaves a sensor device.

My threat model includes attacks from three different sources, each with its own corresponding level of privacy:

1. **Eavesdroppers:** The first level of privacy prevents eavesdroppers from intercepting sensitive data over the radio. Hop-by-hop encryption with symmetric keys [23, 28, 50, 107] is able to achieve this goal.

2. **In-network nodes:** The second level of privacy ensures that individual private information is not disclosed to in-network nodes. These *honest but curious* [75, 105] nodes will follow the network protocols but will mischievously try to learn the sensitive data. This threat model is appropriate because sensors deployed by a common authority can collaborate to fulfill a certain task and it is reasonable that they can be trusted to follow the protocols. This level of privacy is more stringent, but closer to real world situations [124, 144].

3. **Base station:** The third level of privacy ensures that data are not revealed to

anybody, including the final recipient or base station.

I assume that adversaries can capture only a partial number of nodes, information, or packets, or only a partial number of nodes will collude (the amounts of which are later quantified). Finally, I assume adversaries are limited to running in polynomial time based on their input.

Chapter 3

MDNSs: Multi-Dimensional Negative Surveys

The negative surveys described in Chapter 2, Section 3 focus on a single dimension. This chapter extends that preliminary work to multiple dimensions, introducing privacy and utility metrics, a more efficient reconstruction algorithm, and a technique to reduce magnification of error. Extending negative surveys to multiple dimensions increases the range of possible applications, examples of which are given in Chapter 5. Because this technique hides data from every entity, it is well suited to protect human data. Thus, this chapter focuses on participatory sensing applications, in which many users join together to form communities, contributing their sensory information to form a general interactive body of knowledge.

3.1 Introduction

Participatory sensing applications [21] sense, collect, analyze, and share local information collected from a large population of people, enabling a wide range of applications such as urban planning [22], public health [36], and vehicular transportation monitoring [85, 134]. In these applications, the privacy of the people being sensed should be protected, especially when information travels across open wireless networks. On the other hand, there is great social utility in generating high quality data for policymakers, researchers, and the public. Hence, trade-offs exist between the privacy of the participants' data and the utility gained from their content. This trade-off must consider energy efficiency because of the resource-constrained nature of participatory sensing devices.

This chapter applies negative surveys to multivariate categorical data, where categories might be symbolic values (e.g., hair color, race) or a coarse-graining of numerical data into bins. Multi-dimensional data are common in WSNs and participatory sensing applications, which can include several different environmental values along with time and location data. For example, I present a radiation detection scenario in Chapter 5 that determines the distribution of radiation levels at various locations. Participants disguise both dimensions: their geographic location, and their local radiation level. This is important because values from one dimension might inadvertently reveal information about another through correlation analysis. If there are no correlations between sensitive and non-sensitive dimensions, then the non-sensitive values can be reported directly.

Existing approaches for protecting the privacy of multi-dimensional data [5, 69, 113] are designed for database applications, where large numbers of records from different users are available to a centralized server that summarizes statistics about the records [5, 113, 133, 139]. However, in participatory sensing applications, individual participants typically only have access to their own sensed values. They might not be willing to share information with other participants or trust a centralized server to summarize statistics.

One limitation of previous work with negative surveys is the requirement for a large number of participant samples to reconstruct the data accurately [85, 150]. A slight increase in the number of categories requires a significant increase in the number of participants needed to maintain a given level of utility. The problem is compounded when data are multi-dimensional, motivating the work in this chapter. I present a method called dimensional adjustment that reduces error, for a given number of participants. It accomplishes this by sacrificing a small amount of privacy in return for a greater amount of utility, typically 2.5 times more.

Two simulations presented in Chapter 5 illustrate MDNSs. In one, cell phones locate radiation threats such as unexploded dirty bombs, escaped radiation from a nuclear reactor accident, or lost or stolen medical waste, while preserving the privacy of participants' locations. A second simulation reconstructs the underlying probability density function of synthetically generated continuous data, illustrating an alternative approach to random data perturbation (random data perturbation is explained in Chapter 5, Section 1.2).

Abstracting negative surveys to multiple dimensions is not trivial for the following reasons: (1) Different metrics need to be devised to handle multiple dimensions as none currently exist. (2) A method must be devised to manage reconstruction error as the number of dimensions increases. (3) There is little prior work. (4) And, the natural extension of single-dimensional negative surveys (SDNSs) has an exponential time complexity based on the number of dimensions. It is not clear on first inspection that a polynomial time optimization exists.

Chapter Assumptions: The threat model for this chapter includes eavesdroppers listening to radio communications who try to intercept packets, honest but curious intermediate nodes that pass information to the base station, and an honest but curious base station. I assume no data aggregation in the network.

Chapter Contributions: The main contributions of this chapter include: (1) an extension of negative surveys to multivariate categorical data, including a more efficient reconstruction algorithm, (2) privacy and utility metrics for multi-dimensional data, which could be applied to other fields such as privacy-preserving data mining, and (3) a reduction of the needed required participant samples to maintain a given level of utility, given small decreases in privacy, which is also applicable to the single-dimensional case. Chapter 5 extends this work to include: (4) a study of the usability of MDNSs in terms of reconstruction error and the strength of privacy through theoretical analysis and simulations, and (5) a comparison of MDNSs applied to continuous data to randomized data perturbation. Finally, Chapter 6 presents (6) a quantitative comparison of MDNSs to other perturbation approaches.

Chapter Roadmap: The remainder of this chapter is structured as follows. The MDNS protocols are presented in Section 2, followed by Section 3 which describes the privacy and utility metrics used in the analysis. Section 4 discusses the information gained from a MDNS. Dimensional adjustment is introduced and analyzed in Section 5, and Section 6 summarizes this chapter.

3.2 Protocols

Before describing the multi-dimensional node and base station protocols, we introduce some notation. An individual participant senses vector $\vec{x}^{+}$=<x_1^+, x_2^+, ..., x_D^+> from its environment. Real-valued numbers are quantized into categories, if necessary. Each $x_i^+ \in \vec{x}^+$ where $1 \leq i \leq D$, expresses that category x_i was sensed in dimension i. x_i is drawn from a set of categories, C_i={1, 2, $\cdots$, α_i}, that form a proper partition over the data in dimension i, where α_i is the total number of categories for dimension i. The "+" in $\vec{x}^+$ denotes the positive or sensed categorical information, as opposed to the negated or perturbed information represented as $\vec{x}^-$. Subscripts in $\vec{x}_i$ denote the dimension (the i^{th} dimension), while superscripts in $\vec{x}^1$ denote an instance of $\vec{x}$. The collection of participatory sensing application users is known as the *population*. For the entire population, X, Y, and A are D-dimensional matrices which represent the counts by categories of the original, disguised, and reconstructed data sets respectively. For example, if D=3 (i.e. three different environmental variables are sensed) then $X(a,b,c)$, $Y(a,b,c)$, and $A(a,b,c)$ are counts of the number of times the a^{th}, b^{th}, and c^{th} categories appear together in a data set.

3.2.1 Node Protocol

There are three phases to the node protocol:

1. **Sensing:** A node senses a multi-dimensional value $\vec{x}^{+}$ from its environment, course-graining if necessary.

2. **Negation:** For each $x_i^{+} \in \vec{x}^{+}$, the node selects uniformly at random a category x_i^{-} to report to the base station from the set $\{C_i - \{x_i\}\}$, where "$-$" denotes set difference. Hence, $x_i^{-} \neq x_i^{+}$. This is performed independently for each dimension, creating the perturbed vector $\vec{x}^{-}$. The probability of selecting any given category in dimension i is $\frac{1}{\alpha_i - 1}$, where α_i is the number of categories of dimension i. For example in Figure 3.1, a node has sensed $\vec{x}^{+}$=<$2, b$> from its environment, and must choose among the white cells, for instance $\vec{x}^{-}$=<$3, c$>, for a negative value to report back to the base station.

3. **Transmission:** After negation, the node sends $\vec{x}^{-}$ to the base station either immediately, when queried, or according to another protocol.

Pseudocode for the node protocol is given in Algorithm 3.1. Since the number of bits required to transmit either the positive or negative data is identical, there is only a small increase in resources for this phase, due to the cost of obtaining random

	a	b	c	d
1		▨		
2	▨	▨ x	▨	▨
3		▨		

Figure 3.1: Example of positive and negative multi-dimensional space for two dimensions. A sensor that reads $<2, b>$ from its environment selects among the white cells for a value to report to the base station.

Algorithm 3.1 MDNS Node Protocol:

```
for each node $\beta$ do
    procedure Sense
        Sense $\vec{x}^+$ from the environment.
    end procedure
    procedure Negate($\vec{x}^+$)
        for each $\vec{x_i}^+ \in \vec{x}^+$ do
            $\vec{x_i}^- = \text{urand}(C_i - \{\vec{x_i}^+\})$.        ▷ "–" denotes set difference.
        end for
        return $\vec{x}^-$.
    end procedure
    procedure Report($\vec{x}^-$)
        Report $\vec{x}^-$ to the base station immediately or when queried.
    end procedure
end for
```

numbers. Also, because key distribution and management is eliminated, the node protocol saves additional resources [85].

Algorithm 3.2 MDNS Base Station Protocol

```
procedure Base Station Protocol
    Collect all perturbed information, $Y$, from the nodes.
    Compute the estimated distribution, $A$, from the perturbed data, $Y$,
     with Equation (3.1), or the more efficient Algorithm 3.3.
end procedure
```

3.2.2 Base Station Protocol

The base station protocol first collects the reported data, Y, and then estimates the original distributions of sensed values, A, with a reconstruction algorithm. Pseudocode is given in Algorithm 3.2. Since the protocol is straightforward, I focus on the reconstruction algorithm in the following. First, I introduce a natural multi-dimensional extension to the single-dimensional equation and show that is has exponential time complexity, then I present a time optimization, and finally give an algorithmic simplification.

Natural Extension of SDNSs to MDNSs

Single-dimensional negative surveys use Equation (2.4) [63, 85] to estimate the original distribution. A natural extension to D dimensions is:

$$\forall \vec{x} \mid A(\vec{x}) = N + \sum_{k=1}^{D} (-1)^k \cdot \Gamma(\vec{x}, k), \tag{3.1}$$

where $\Gamma(\vec{x}, k)$ is given as:

$$\Gamma(\vec{x}, k) = \sum_{\substack{d \in \\ B(\{1,\ldots,D\},k)}} \left(\left[\prod_{j \in d} (\alpha_j - 1) \right] \cdot \sum_{\substack{\vec{y}\ s.t. \\ y_i \in \vec{x}, \\ \forall i \in d}} Y(\vec{y}) \right), \tag{3.2}$$

and $B(\{1,\dots,D\},k)$ are all the k length possible combinations of members of {1, ..., D}. For example, $B(\{1,2,3\},2)$ is {{1,2},{1,3},{2,3}}. $Y(\vec{x})$ is the count of the reported disguised sensed values that have categories specified by d from $\vec{x}$. Each dimension must use a NSPM. As an example, Equation (3.1) with D=3 is given as:

$$\begin{aligned}\forall a,b,c \mid A(a,b,c) = \sum_{\vec{x}} Y(\vec{x}) - (\alpha_1{-}1)\sum_{\substack{\vec{x}\ s.t.\\ x_1=a}} Y(\vec{x}) - (\alpha_2{-}1)\sum_{\substack{\vec{x}\ s.t.\\ x_2=b}} Y(\vec{x})\\ -(\alpha_3{-}1)\sum_{\substack{\vec{x}\ s.t.\\ x_3=c}} Y(\vec{x}) + (\alpha_1{-}1)(\alpha_2{-}1)\sum_{\substack{\vec{x}\ s.t.\\ x_1=a,\\ x_2=b}} Y(\vec{x}) + (\alpha_1{-}1)(\alpha_3{-}1)\sum_{\substack{\vec{x}\ s.t.,\\ x_1=a,\\ x_3=c}} Y(\vec{x})\\ +(\alpha_2{-}1)(\alpha_3{-}1)\sum_{\substack{\vec{x}\ s.t.\\ x_2=b,\\ x_3=c}} Y(\vec{x}) - (\alpha_1{-}1)(\alpha_2{-}1)(\alpha_3{-}1)\sum_{\substack{\vec{x}\ s.t.\\ x_1=a,\\ x_2=b,\\ x_3=c}} Y(\vec{x}).\end{aligned} \tag{3.3}$$

The time complexity of Equation (3.1) is given as:

$$O\left(\left[\prod_{i=1}^{D}\alpha_i\right]\cdot\left[\sum_{i=1}^{D}\binom{D}{i}(D-i)\cdot\alpha_{max}\right]\right). \tag{3.4}$$

where α_{max} is the maximum number of categories among the dimensions. There are $\sum_{i=1}^{D}\binom{D}{i}$ total Y terms that require $(D-i)$ calculations. α_{max} guarantees that enough calculations are accounted for. Since $\binom{D}{0}+\binom{D}{1}+\cdots+\binom{D}{D}=2^D$, this equation is exponential with respect to the number of dimensions.

Time Optimization: Matrix Memoization

The reconstruction algorithm shown in Equation (3.1) can be improved with the more time efficient scheme presented in Algorithm 3.3. This algorithm uses *matrix memoization* to improve the running time and generalizes to any perturbation matrix, not just a NSPM. The inputs to Algorithm 3.3 are: (1) D, the number of dimensions, (2) Y, the D-dimensional matrix of disguised values, (3) $F = [\alpha_1,\dots,\alpha_D]$, a list of the number of categories for each dimension, and (4) $M = [M_1,\dots,M_D]$, the perturbation matrices for each dimension. The symbol ":" denotes a slice operator, an operation on a matrix designating every element in the dimension in which it appears [1]; τ is a function similar to transpose that takes a row, column, hyper-row, or hyper-column, and transforms it into a vector appropriate for matrix multiplication. *index* is constructed to be a vector of length D, with one member of the vector consisting of ":". When used as an index into R, it returns a vector.

The time complexity of Algorithm 3.3 is:

$$O\left(\sum_{i=1}^{D}\left[\prod_{j=1,j\neq i}^{D}\alpha_i^2\alpha_j\right]\right) = O\left(\sum_{i=1}^{D}\alpha_i^2\cdot\prod_{i=1}^{D}\alpha_i\right), \tag{3.5}$$

ignoring the cost of matrix inversion for each M_δ. Intuitively, the complexity arises from the matrix multiplication with every possible vector in R. Each update of R from Line 11 in Algorithm 3.3 stores information back in R for other overlapping vectors to use, thus reducing the total amount of computation. Only one dimension in R can be updated at a time or the algorithm will produce inaccurate results. The technique also works for other perturbation matrices (e.g. RRTs). However, when a NSPM is used for each dimensional perturbation matrix, the cost of Algorithm 3.3 reduces to:

$$O\left(D\cdot\prod_{i=1}^{D}\alpha_i\right), \tag{3.6}$$

because Line 11 in Algorithm 3.3 is replaced with the simpler Equation (2.4). Equation (3.6) is clearly an improvement over Equation (3.4).

[1] The part of Algorithm 3.3 that handles the slice operator and *index* variable was designed by Benjamin Edwards, however the original algorithm to update each vector and copy it back into R was designed by the author in a slightly different formulation.

Algorithm 3.3 Reconstruction Optimization for D Dimensions

```
function RECONSTRUCT_MATRIX($Y, D, F, M$)
    $R = Y$
    for $\delta \in [1 : D]$ do
        update_dim($R, D, [\ ], \delta, F, M$)
    end for
    return $R$
end function

function UPDATE_DIM($R, D, index, \delta, F, M$)
    if $length(index) = D$ then
        $R(index) \leftarrow M_\delta^{-1} * R(index)^\tau$
    else if $len(index) + 1 = \delta$ then
        $new\_index \leftarrow index.append([:])$
        update_dim($R, D, new\_index, \delta, F, M$)
    else
        for $i \in [1 : F(length(index) + 1)]$ do
            $new\_index \leftarrow index.append([i])$
            update_dim($R, D, new\_index, \delta, F, M$)
        end for
    end if
end function
```

Algorithmic Simplification

The *Kronecker technique* converts a MDNS to a single dimension. Although the complexity cost is greater than Equation (3.6), this simplifies implementation and allows the use of single-dimensional metrics, which produce the same values as their multi-dimensional counterparts.

The Kronecker technique uses a perturbation matrix, M', that is the *Kronecker product* [86] of the individual perturbation matrices for each dimension, given as:

$$M' = (((M_1 \otimes M_2) \otimes M_3) \ldots \otimes M_D), \tag{3.7}$$

where $\otimes$ is the Kronecker product operator. The Kronecker product of two matrices is the tensor product with respect to a standard choice of basis [86]. Y is transformed into a new vector Y', an $n \times 1$ vector where n is the product of the number of categories in each dimension. For example, if Y has three dimensions with 4, 3, and 2 categories each, Y' is given as:

$$Y' = \begin{bmatrix} Y(1,1,1) \\ Y(1,1,2) \\ Y(1,2,1) \\ Y(1,2,2) \\ Y(1,3,1) \\ \vdots \\ Y(4,3,2) \end{bmatrix}. \tag{3.8}$$

To obtain the estimated distribution, A, Y is multiplied with $(M')^{-1}$ according to Equation (2.3). A is then transformed to a D-dimensional matrix, taking care that the order of transformations corresponds to the order that the Kronecker products were applied to the perturbation matrices. Although convenient, this technique is not optimal because of the time complexity which is given as:

$$O\left(\left[\prod_{i=1}^{D} \alpha_i\right]^2\right), \tag{3.9}$$

ignoring matrix inversion which is minimal because of the mixed-product property [86].

3.3 Privacy and Utility Metrics

Using privacy and utility metrics extended from Huang and Du [89], I quantify the trade-offs between the accuracy of reconstruction and the amount of privacy protected. Like their single-dimensional counterpart, the multi-dimensional formulations apply to any perturbation matrix, not necessarily a NSPM. The privacy metric ranges from [0,1], while the utility metric ranges from $[0, +\infty)$. For both metrics lower values are desirable. These privacy and utility metrics, and some terminology, are borrowed from the privacy preserving data mining (PPDM) field. The terms accuracy, reconstruction error, and utility are used interchangeably, as are the terms disguise, perturb, and negate.

3.3.1 Privacy Metric

The privacy metric measures the probability of guessing the original data from the disguised values, and is based on the *maximum a posteriori* (MAP) estimate. Huang and Du [89] theorize that the MAP estimate is the "best that adversaries can achieve when their estimation is consistent," and it gives an upper bound on an adversary's threat. I extend their single dimensional metric to multiple dimensions as follows:

$$Privacy = \sum_{\substack{\Upsilon \in Y(\vec{x}) \\ \forall \vec{x}}} P(\Upsilon | \widehat{X_\Phi}) \cdot P(\widehat{X_\Phi}), \tag{3.10}$$

where

$$\widehat{X_\Phi} = \arg \max_{\substack{\Phi \in X(\vec{x}) \\ \forall \vec{x}}} P(\Phi | Y). \tag{3.11}$$

Equation (3.11) calculates for Equation (3.10) the optimal MAP estimate for a given index, $\vec{x}$, of Y (the maximum index, $\vec{x}$, in each column of $P(X|Y)$).

If an adversary has no prior knowledge of the underlying distribution, I propose that privacy generalizes to k-indistinguishability. I define an item to be k-indistinguishable if it cannot be identified with higher probability than guessing from $k-1$ other items. A participant's reported data in a SDNS with α categories has a k-indistinguishability value of $\alpha - 1$. An individual's data in a MDNS with categories α_1, $\alpha_2, \ldots,$ α_D will have a k-indistinguishability value of $(\alpha_1 - 1) \cdot (\alpha_2 - 1) \cdot \ldots \cdot (\alpha_D - 1)$. This is different from k-anonymity in WSNs [6, 80, 113, 133, 139] which preserves location information and measures the ability of an adversary to distinguish a participant from a set of $k - 1$ nearby participants.

3.3.2 Utility Metric

Utility, also known as accuracy or reconstruction error, measures the difference between the original, X, and reconstructed, A, data distributions. I use the following reasoning from Huang and Du [89]. Since A is an unbiased maximum likelihood estimate of X, the mean of the estimate A is identical to the original distribution X. Yet, each specific estimate A deviates from X by some amount. The closer A is to X, the higher A's utility. Hence, the mean square error (MSE), given as follows, is used to quantify utility:

$$MSE = E[(A - X)^2]. \tag{3.12}$$

Huang and Du [89] actualized this equation by replacing X with the mean of A to estimate A's variance. Using Equation (2.3), they equate the variance of A and $M^{-1}Y$, and state a theorem to compute the MSE. I extend this theorem to multiple dimensions with the following equation:

$$\begin{aligned} Utility &= \frac{1}{\alpha_1 \cdot \ldots \cdot \alpha_D} \sum_{\vec{x}^i} MSE(X = \vec{x}^i) \\ &= \frac{1}{\alpha_1 \cdot \ldots \cdot \alpha_D} \sum_{\vec{x}^i} E[(P(A = \vec{x}^i) - P(X = \vec{x}^i))^2] \\ &= \frac{1}{\alpha_1 \cdot \ldots \cdot \alpha_D} \sum_{\vec{x}^i} \Bigg(\sum_{\vec{x}^j} \Big[\mu(\vec{x}^i, \vec{x}^j)^2 \cdot var(\vec{x}^j) \Big] \\ &\quad + \sum_{\substack{\vec{x}^k, \vec{x}^\ell \; s.t. \\ \vec{x}^k_\gamma \neq \vec{x}^\ell_\gamma, \; \forall \gamma}} \Big[2 \cdot \mu(\vec{x}^i, \vec{x}^k) \cdot \mu(\vec{x}^i, \vec{x}^\ell) \cdot cov(\vec{x}^k, \vec{x}^\ell) \Big] \Bigg), \end{aligned} \tag{3.13}$$

where

$$\mu(\vec{x}^m, \vec{x}^n) = \prod_{d=1}^{D} M_d^{-1}(\vec{x}^m_d, \vec{x}^n_d), \tag{3.14}$$

denotes the product of the elements from the inverse of the perturbation matrix for each dimension where the row and column correspond to the categories in the d^{th} dimension of $\vec{x}^m$ and $\vec{x}^n$ respectively. *var* and *cov* are given as:

$$\begin{aligned} var(\vec{x}^i) &= \frac{1}{N} \cdot P(Y = \vec{x}^i) \cdot (1 - P(Y = \vec{x}^i)) \\ cov(\vec{x}^i, \vec{x}^j) &= -\frac{1}{N} \cdot P(Y = \vec{x}^i) \cdot P(Y = \vec{x}^j), \end{aligned} \tag{3.15}$$

The actual variance and covariance of a MDNS are given as:

$$\begin{aligned} var_{MDNS}(\vec{x}^i) &= \frac{\left(\left[\prod_{i=1}^{D} \alpha_i \right] - 1 \right)^2}{N} \cdot P(Y = \vec{x}^i) \cdot (1 - P(Y = \vec{x}^i)) \\ cov_{MDNS}(\vec{x}^i, \vec{x}^j) &= -\frac{\left(\left[\prod_{i=1}^{D} \alpha_i \right] - 1 \right)^2}{N} \cdot P(Y = \vec{x}^i) \cdot P(Y = \vec{x}^j), \end{aligned} \tag{3.16}$$

and were verified using the Algorithmic Simplification in Section 2.2. When an MDNS is converted to a single dimension, Equation (3.16) is equal to the single-dimensional variance and covariance equations used in Esponda and Guerrero [63].

3.3.3 Experimental Study of Trade-offs Between Privacy and Utility

The underlying distribution, X, affects utility and privacy. I use the following normalized version of Shannon's entropy to illustrate the effects:

$$S = \frac{-\sum\limits_{\vec{x}} P(X = \vec{x}) \log P(X = \vec{x})}{\log\left(\prod\limits_{i=1}^{D} \alpha_i\right)}, \tag{3.17}$$

where S is in $[0, 1]$. For example, a spiked distribution (all elements in one category) has the lowest normalized entropy, $S = 0$, and provides the worst privacy, but the highest utility. A uniform distribution, which has the highest normalized entropy, $S = 1$, provides the worst utility, but the best privacy. All other distributions fall between these two extremes. However, the underlying distribution affects privacy significantly more than utility. For example in Figure 3.2, the spiked and uniform distributions span 87.4% of the entire privacy metric. These two distributions span a significantly smaller range of utility, 0.786 to 0.802, which corresponds to 1.6% of the privacy metric. Since this effect on utility is so small, Groat et al. [78] interpreted utility to be independent of the underlying distribution. This is a reasonable simplification because the number of categories and the number participants dominate the metric's value. It has the advantage of allowing WSN designers to determine the utility of a negative survey without knowing the distribution of the original data.

3.4 Analysis of Adversarial Information Gained

The amount of information gained about the original sensed value by an adversary who intercepts a response from a node in a SDNS was proved to be less than or equal to what is gained with a positive survey by Esponda et al. [63]. Following this logic,

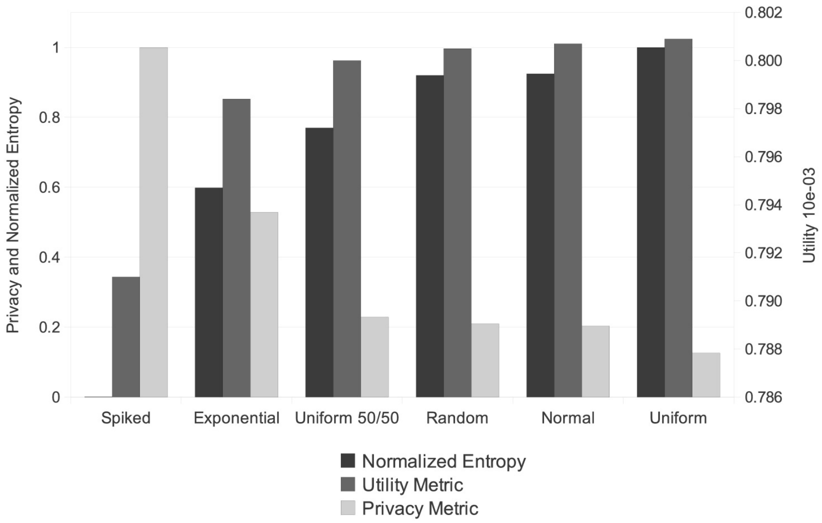


Figure 3.2: Effects of different original distributions on privacy and utility.

the amount of information that can be gained from a MDNS is formalized as the information gained from a positive survey minus the information gained from the same survey where $\vec{x}^{s-}$ has been removed. $\vec{x}^{s-}$ is the negative data a node transmits to the base station that contains one perturbed category from each dimension. The information gained by an adversary that intercepts $\vec{x}^{s-}$ is formalized for a D-dimensional MDNS as follows:

$$\begin{aligned}
& I(< i, j, ..., k >, \forall\ i, j, ..., k | X \neq \vec{x}^{s-}) = -\sum_{\ell=1}^{\alpha_1} \sum_{m=1}^{\alpha_2} ... \sum_{n=1}^{\alpha_D} P(X =< \ell, m, ..., n >) \\
& \cdot \log P(X =< \ell, m, ..., n >) \\
& + \sum_{\substack{\ell = 1 \\ s.t.\ \ell \neq \vec{x}_1^{s-}}}^{\alpha_1} \quad \sum_{\substack{m = 1 \\ s.t.\ m \neq \vec{x}_2^{s-}}}^{\alpha_2} \quad ... \quad \sum_{\substack{n = 1 \\ s.t.\ n \neq \vec{x}_D^{s-}}}^{\alpha_D} \quad P(X =< \ell, m, ..., n > | X \neq \vec{x}^{s-}) \\
& \cdot \log P(X =< \ell, m, ..., n > | X \neq \vec{x}^{s-}). \qquad (3.18)
\end{aligned}$$

where $<i, j, ..., k>$ denotes a sample that has category i in the first dimension, category j in the second dimension, and category k in the D^{th} dimension. The probabilities in Equation (3.18) reflect the distributions of the original environment.

Equation (3.18) reports the information gained by an adversary if one negative response is collected. The following discusses the possibility of an adversary collecting many negative responses while the original sensed value remains the same. Eventually, after receiving all possible negative samples, the adversary will have gained all information about the original sensed value. Assuming a uniform distribution for X, Figure 3.3 illustrates this concept. On the left is a SDNS of 200 categories. When 199 unique categories have been received by the adversary, it has gained 7.6439 bits of information about the original value, $2^{7.6439} = 200$. On the right, an adversary receives information in 2 dimensions. It is possible that all categories except the sensed category could be seen in the first dimension while only one category is seen in the second dimension. This correlates to the front right corner in the surface plot. The amount of information gain when all negative responses have been seen is also 7.6439, because there are 200 possible responses (10 by 20 categories).

3.5 Dimensional Adjustment Improves Efficiency

In this section, I introduce a technique called *dimensional adjustment (DA)* that reduces the number of participants required to obtain reasonable utility. It accomplishes this by constructing extra dimensions while maintaining the total number of categories. DA addresses a limitation of the previous work on single-dimensional data [85, 150], namely, that as the number of categories increases for a given dimension, many additional samples are required to maintain a constant utility value. This limitation is compounded in multiple dimensions, potentially limiting negative surveys to applications with a small number of categories. I propose DA to address this challenge, discussing the privacy and utility trade-offs, explaining the magnification of error, and illustrating that it always improves utility and is expected to reduce

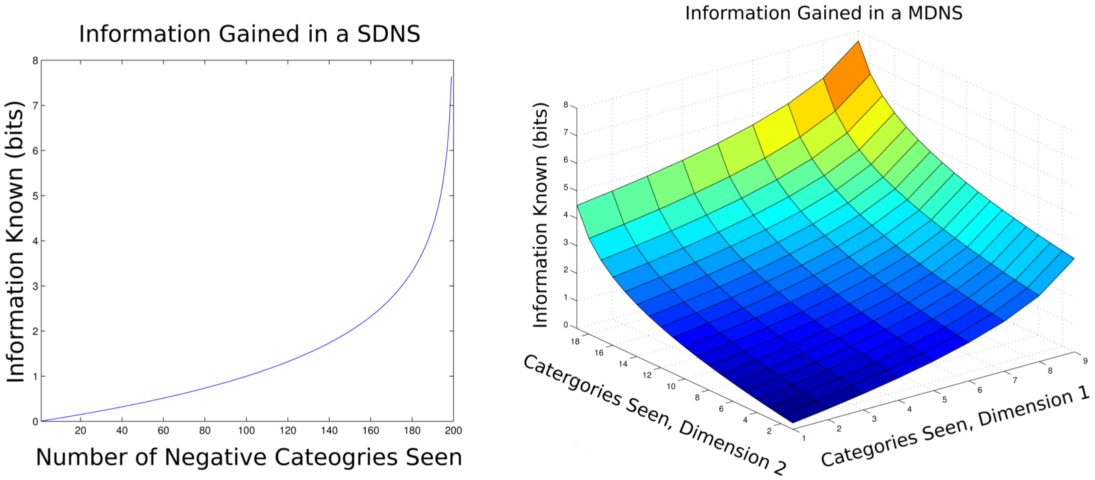


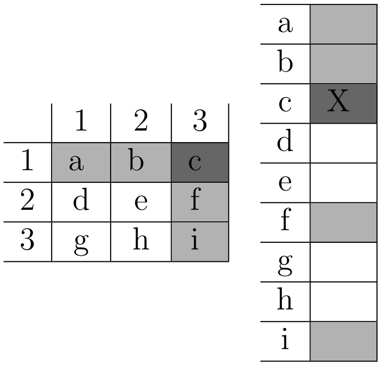
Figure 3.3: (Left) The amount of information gained by an adversary in a 200 category SDNS who captures negative responses from a node, assuming the positive value remains the same. (Right) The amount of information gained in a 20 by 10 MDNS by an adversary that captures negative responses, assuming the positive information remains the same. The maximum value in either graph is 7.6439, the number of bits needed to represent 200 categories.

privacy.

3.5.1 Dimensional Adjustment Algorithm

DA distributes a fixed number of categories into extra dimensions. For example, a one-dimensional negative survey containing 64 categories can be remapped to: 2 dimensions of 8 categories each, 2 dimensions of 4 and 16 categories, or any number of dimensions where the product of the number of categories in each dimension equals 64. Remapping dimensions is easy to implement and is similar to base conversion with variable bases, as illustrated in Figure 3.4.

Splitting data into multiple dimensions with a smaller number of categories for each dimension improves reconstruction accuracy (utility). Fewer dimensions with a



	1	2	3
1	a	b	c
2	d	e	f
3	g	h	i

a	▨
b	▨
c	X
d	
e	
f	▨
g	
h	
i	▨

Figure 3.4: An example of dimensional adjustment where the alphabetic dimension is adjusted to two dimensions of three categories each. A node that senses <c> must choose among the white cells to report.

larger number of categories worsens utility (higher utility value). Intuitively, accuracy is related to Figure 3.1 and the ratio of the white squares (negative information) to the total number of squares. Figure 3.4 illustrates DA when a single dimension containing 9 categories is reduced to two dimensions of 3 categories each. As the number of dimensions increases, and the number of distinct categories remains constant, this ratio decreases, reducing the number of possible perturbations, which increases accuracy of reconstruction. The next section analyzes these trade-offs.

3.5.2 Trade-off Analysis

Using DA to transform a low-dimensional survey into a high-dimensional survey involves trade-offs. For example, a one-dimensional negative survey with 64 categories provides the highest privacy but the lowest utility. However, when the same data are mapped to 6 dimensions with 2 categories each, the reconstruction provides the lowest privacy but the best utility. The relationship between privacy and utility is nonlinear, providing an opportunity to optimize. For example, in Table 3.1 with 1,000,000 samples and 10,000 categories, privacy degrades 35% while utility improves

Table 3.1: Two negative surveys of 10,000 total categories and 1,000,000 participants. The second uses dimensional adjustment.

	1 dimension of 10,000 categories	6 dimensions of 5x5x5x5x4x4 categories
utility	0.00100	0.00014
privacy	0.01457	0.01960

86%, where percentage is calculated as:

$$\frac{y - x}{x}. \tag{3.19}$$

Using Table 3.1 and the following estimate equations for privacy and utility, I further illustrate these trade-offs. Without loss of generality, the original distribution, X, is assumed to be normal. I use a simple empirically discovered linear model to estimate utility for a SDNS given N participants and α categories:

$$Utility_{Estimate} = \frac{(\alpha - 2)}{N}, \tag{3.20}$$

which has an R^2 value of 0.9999 when either N or α varies. I estimate privacy as a function of the number of categories in a SDNS as:

$$Privacy_{Estimate} = \frac{2.5}{(\log_2(\alpha))^2 + 1.5}, \tag{3.21}$$

which has $R^2 = 0.976$. While the utility estimate does not depend on the original distribution, the privacy estimate does. The empirically discovered Equation (3.21) will only estimate accurately the privacy of a reconstructed normal distribution.

Using Equations (3.20) and (3.21), I can estimate the population size, or number of categories required to achieve a target utility or privacy value. For the data in Table 3.1, Equation (3.20) shows that one dimension of 10,000 categories and 71,414,286 participants is equivalent to 6 dimensions (where 4 dimensions have 5 categories and

2 dimensions have 4 categories) with 1,000,000 participants. The MDNS requires fewer participants. When participants are fixed at 1,000,000, Equation (3.20) also indicates that a MDNS of 10,000 categories using DA is equivalent to using 142 categories in a SDNS. Equation (3.21) indicates that when the population is fixed at 1,000,000, a MDNS of 6 dimensions and 10,000 overall number of categories is equivalent in privacy to a SDNS using 2,397 categories. Since privacy degrades and utility improves when the number of categories decreases, the above information indicates a privacy degradation of 70.0% but a utility improvement of 98.58%. These percentages will not linearly correlate with the privacy and utility metrics, yet they do show how the privacy-utility trade-off is favorable for DA.

3.5.3 Magnification of Error

Previous works [85, 150] have suggested reasons why an increase in categories requires a significant increase in participants to maintain a given level of utility. Horey et al. [85] suggest that an almost linear increase in the number of participants is needed to maintain a given utility as the number of categories increase, when utility is measured with the relative Root Mean Square Error (RMSE). When utility is measured with the MSE, as illustrated in Figure 3.5, the relation is indeed linear. Xie et al. [150] suggests that the magnification of error is due to counting with integers, pointing to a gap between the floor and ceiling of the value $(X_i)/(\alpha - 1)$. This gap introduces errors in the reconstruction process which are increased with the total number of categories. They give an upper and lower bound of the error for a given category to be $\pm(\alpha - 1)^2$. However, they also assume that the RMSE is the measure of utility. Utility metrics based on the RMSE or MSE are misleading because they do not model how the perturbed distribution, Y, deviates from its expected value. Values in Y that are closer to their expected values give a better reconstructed distribution, A, that is closer to the original data distribution, X.

I introduce an alternative explanation using Chernoff bounds which better estimate the deviation of Y from its expected value. Since negative surveys are similar to the balls and bins problem [24], its notation (balls are sensed values and bins are categories) will be used for the rest of this section. In a negative survey with an original distribution, X, the balls in category X_i must be distributed among the other $\alpha - 1$ bins. This series of Bernoulli trials is binomially distributed. Chernoff bounds approximate the binomial distribution and are especially good for representing the tails far from the mean. The expected number of balls in the disguised bins, Y, is calculated by first taking the inverse function of Equation (2.4) as follows:

$$E[Y_i] = \frac{N - A_i}{\alpha - 1}. \tag{3.22}$$

Since this is a maximum likelihood estimate, A_i can be replaced with X_i. This equation and the following two assume a SDNS, but a MDNS would behave similarly, if α in Equations (3.22) and (3.23) is replaced with all the possible categories a sensed value could be perturbed to, i.e., α is replaced with $(\alpha_1 - 1) \cdot (\alpha_2 - 1) \cdot \ldots (\alpha_D - 1)$.

The Chernoff upper and lower bounds, which determine the probability that a bin will be filled with δ more or less balls than the expected value, are represented as follows:

$$\begin{aligned} P[X_i > E[Y] + \delta] &= \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right)^{\frac{N-X_i}{\alpha-1}} \\ P[X_i < E[Y] - \delta] &= \left(\frac{e^{\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\frac{N-X_i}{\alpha-1}}. \end{aligned} \tag{3.23}$$

Without loss of generality (and for graphing), δ is fixed at one normalized standard deviation of the binomial distribution given as:

$$\delta = \sqrt{\left(\frac{1}{\alpha}\right) \cdot \left(1 - \frac{1}{\alpha}\right)}. \tag{3.24}$$

Figure 3.5 gives the results of the Chernoff bounds when there are 10 categories and the population varies, and then when the population is fixed at 1,000 and the number of categories varies. In the figure (left), when the population increases, the probability of a bin in Y being filled with more balls than one standard deviation from its expected value decreases. If the y-axis is scaled logarithmically, the Chernoff bounds form a straight line, suggesting that an increase in participants exponentially (with an exponent of -0.70 in Figure 3.5 left) decreases the deviation of a bin in Y from its expected value. Figure 3.5 (right) shows that when the number of categories increases, the probability of a bin deviating from its expected value grows almost logarithmically. To maintain a constant probability of deviation error for a given change in the number of categories, the population is increased as follows:

$$N_{increase} = \alpha_{\Delta} \cdot log(N), \tag{3.25}$$

where α_{Δ} is the increase in the number of categories, and N is the original population. In this equation, the required number of participants increases linearly with the number of categories. Any base can be used for the log, however, lower values give lower Chernoff bound probabilities.

In Figure 3.5 (right), the probability of deviating from the expected value grows almost logarithmically with an increase in categories, however, the initial increase (from 3 to 150 categories) is significant. This significance increase of the Chernoff bounds could explain the magnification of error associated with negative surveys. As the number of categories increases, the values in Y have a higher probability of deviating more than one standard deviation from the expected values. While the utility metric uses a different unit of measurement, when compared to the Chernoff upper and lower bounds, the metric underestimates the reconstruction error.

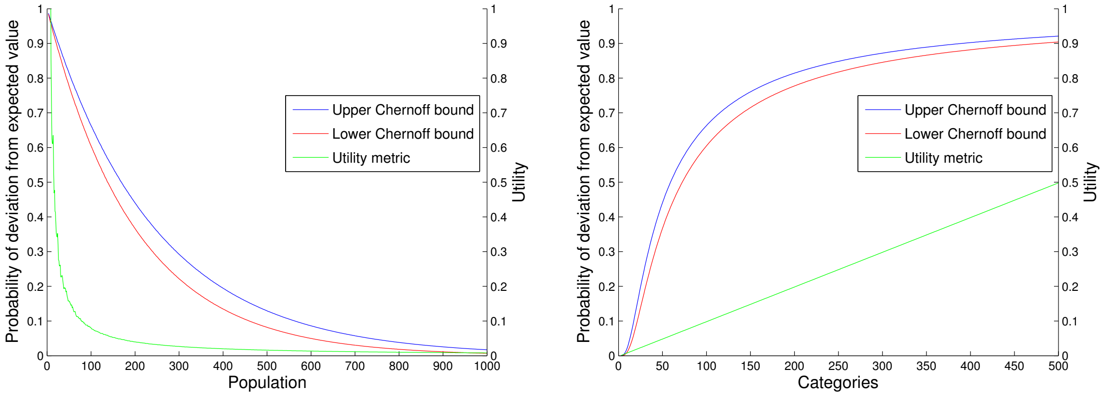


Figure 3.5: Chernoff upper and lower bounds showing the probability that the values in Y deviate from their expected values. The number of categories are fixed at 10 and the population varies (left), and population is fixed at 1,000 and the number of categories varies (right). The utility metric (green) is included, but uses a different unit of measurement. However, compared to the Chernoff upper and lower bounds, the metric underestimates the reconstruction error.

3.5.4 Dimensional Adjustment Always Improves Utility

In this section I argue that DA always improves utility. Assuming that no negative survey has fewer than three categories[2], the limit of the Chernoff upper bound as α approaches three from the right is:

$$\lim_{\alpha\to 3^+}\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\frac{N-X_i}{\alpha-1}} = \left(e^{\delta}(\delta+1)^{(-\delta-1)}\right)^{\frac{N-X_i}{2}}. \tag{3.26}$$

The derivative of the Chernoff upper bound is given as:

$$\frac{\partial}{\partial\alpha}\left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\frac{N-X_i}{\alpha-1}} = -\frac{\log(e^{\delta}(\delta+1)^{-\delta-1})(N-X_i)(e^{\delta}(\delta+1)^{-\delta-1})^{\frac{N-X_i}{\alpha-1}}}{(\alpha-1)^2} \tag{3.27}$$

which is always positive when α, N, δ, X_i are greater than zero, and $\delta<1$ and $X_i<N$. This indicates that the upper bound in Equation (3.23) is monotonically increasing

[2] A survey with 2 categories is simply the bitwise inverse of the data and provides no privacy, and a survey with 1 category is not very interesting.

as α increases (or monotonically decreasing as α decreases). DA always reduces the overall number of possible categories a value can be perturbed to. For instance, if a dimension denoted as k has α_k categories where $\alpha_k = \prod_{i=1}^{D_k} \alpha_i$, then the number of available categories when the dimension is dimensionally adjusted to D_k dimensions is $(\alpha_{k,1} - 1) \cdot (\alpha_{k,2} - 1) \cdot \ldots \cdot (\alpha_{k,D_k} - 1)$. Since $\alpha_k > (\alpha_{k,1} - 1) \cdot (\alpha_{k,2} - 1) \cdot \ldots \cdot (\alpha_{k,D_k} - 1)$, the number of possible categories to perturb to are always reduced. This will always tighten the distribution of values for each bin in Y, i.e., the bins will be closer to their expected amount. This improves the reconstructed distribution (A values are closer to X), which improves utility.

Additionally, as the population size goes to infinity, the limit of the Chernoff bound approaches 0, as given below:

$$\lim_{N \to \infty} \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}} \right)^{\frac{N - X_i}{\alpha - 1}} = 0. \tag{3.28}$$

This illustrates how reconstruction is a maximum likelihood estimate. More participants gives values in Y closer to their expected values.

3.5.5 Dimensional Adjustment is Expected to Reduce Privacy

Without loss of generality, any MDNS can be adjusted to a single-dimensional negative survey (see Section 2.2 in this chapter). When examining the joint probability of X and Y in this case, some cells have zero entries because they are invalid perturbations (i.e., the original and perturbed values share at least one value). When a SDNS or a dimension in a MDNS is dimensionally adjusted, zeros are added to various cells in the joint probability. However,the other cells in the joint probability are expected to increase, especially the maximum values in the columns. This will increase the value of Equation (3.11), which will also increase the value of Equation (3.10), reducing privacy. While the privacy metric is not guaranteed to increase, it is expected to increase, especially with large populations.

3.6 Summary of Chapter

Information can be collected through MDNSs of participants' devices. Participants report false information to preserve their privacy. However, the original distributions across the dimensions can be reconstructed from this false information. In this chapter, I introduced the MDNS protocols to collect and reconstruct such information. An efficient reconstruction algorithm was devised along with metrics for multiple dimensions. A technique was put forward to reduce the inherent need for a large number of participants. In the next chapter, I discuss another information collection protocol that also preserves privacy and reduces energy use, but performs data aggregation on the information.

Chapter 4

KIPDA: *k*-Indistinguishable Privacy-preserving Data Aggregation

The previous chapter discussed privacy-preserving data transmission without aggregation in the network. Although data aggregation could be performed with MDNSs, a separate value would have to be transmitted for each distinct category. In a MDNS with many dimensions, this would defeat the purpose of saving energy. This chapter presents KIPDA, a lightweight k-indistinguishable CDA algorithm for the maximum (MAX) and minimum (MIN) aggregation functions. Instead of perturbing information as in the previous chapter, the data values are hidden in plain site among camouflage values.

4.1 Introduction

WSNs often combine, process, or filter data between the sensor and the final destination, a process previously described as data aggregation. This chapter is concerned with the problem of maintaining privacy when the data are aggregated in the network. I introduce a non-cryptographic method called *KIPDA*, or *k-indistinguishable privacy-preserving data aggregation*, which obfuscates data by adding a set of camouflage values. In KIPDA, the aggregates, referred to as the *sensitive values*, are transmitted in plaintext so that the aggregation computation is efficient. These sensitive values are disguised with camouflage values in a *message vector*, a one-dimensional array of values which is defined as the union of the sensitive value with the camouflage values. WSN nodes transmit message vectors to their parents in the data aggregation tree, typically in a single packet. k-Indistinguishability of the sensitive values from the camouflage data is achieved by choosing the values and positions of the camouflage data in the message vector in such a way that the sensitive values are aggregated correctly and can be decoded at the final destination or base station.

The technique is counter-intuitive, because it takes extra energy to transmit camouflage values. However, I show through analysis, and in Chapter 5 with simulations, that KIPDA is more energy efficient than using end-to-end data collection without aggregation, or hop-by-hop aggregation with five current conventional encryption ciphers. KIPDA also excels at timing, and can aggregate and transmit significantly more decoy values than hop-by-hop aggregation can aggregate and transmit one sensitive value. This could be important in delay intolerant networks [129].

KIPDA is one of the first MAX/MIN CDA techniques to protect information from in-network nodes with energy-efficiency. It is robust to nodes that are captured and re-programed to follow network protocols (so as to appear benign) to determine sensitive data. The fewer nodes that are controlled by an adversary, the less ability the adversary has to distinguish sensitive values from camouflage values. This is accomplished by using a method similar to a global symmetric key [99], except that each node possesses a random part of the global key. Only when enough nodes collude or are captured will privacy be broken. Neither hop-by-hop encryption aggregation or end-to-end aggregation with homomorphic encryption support MAX/MIN aggregation with protection from in-network nodes (see Chapter 2, Section 2).

CDA MAX/MIN aggregation is very similar to secure multi-party computation, which is a general case of the millionaire problem, where two people want to know who is richer without revealing their true wealth. The only amount of information gained is from the result of the answer. The solution to this problem is too resource expensive to be applied to WSNs. For example, previous solutions to Yao's Millionaire Problem [40, 152] leverage public-key cryptography, which is computationally expensive and therefore problematic in resource-constrained WSNs.

Several applications could benefit from KIPDA and MAX/MIN aggregation. For example, intelligent or smart meters for electric utilities send individual usage data to a utility company, which then sends real-time data back to the end user to encourage energy conservation [77]. Information from the meter is usually sent over an existing cell phone infrastructure, radio transmission, or other unsecured network. Privacy is essential in this setting, least others can infer daily activities by observing utility consumption patterns [77]. Another potential application arises in medicine, if a medical worker does not have the time or resources to monitor a large group of patients individually. Determining the MAX or MIN value of an indicator could show that the entire group is within the normal range, or that a patient is in trouble and needs attention. Without privacy-preserving techniques, an eavesdropper could observe values of a patient's health data. A similar idea could be used to triage patients at a disaster site [82].

Chapter Contributions: The contributions of this chapter include the following:

(1) a MAX/MIN CDA scheme that protects data from in-network nodes, (2) analysis of the ability of an adversary to distinguish sensitive data from camouflage data as a function of the number of nodes she has captured, (3) a detailed analysis of the energy consumption and time delay of KIPDA compared to end-to-end data collection and hop-by-hop aggregation encryption schemes. In Chapter 5, this energy and time analysis is compared to the results from a power aware WSN simulator (PowerTOSSIM-Z) and actual implementations on physical devices.

Chapter Assumptions: I assume in-network nodes that transmit message vectors are honest but curious, along with the base station. Communications from each sensor node to the base station can also be monitored by an adversary. Communications from the base station to each node is performed securely to distribute necessarily information about the positions of the camouflage data in a node's message vector.

Chapter Roadmap: The remainder of this chapter is organized as follows: Section 2 provides an overview of the protocols. Section 3 discusses aggregation accuracy, and analyzes the abilities of an adversary through node capture and collusion attacks. Section 4 presents the protocols in detail. Section 5 analyzes the power consumption of KIPDA and compares it to similar techniques, and Section 6 summarizes this chapter.

4.2 Overview of Solution

Before I present the overview of my solution, I introduce notation which is summarized in Table 4.1 and illustrated in Figure 4.1. Let V^i be the set of n values in a message vector for node i where $(|V^i| = n,\ \forall i,\ 1{\le}i{\le}N)$, and where N is the number of participating nodes in the network. The message vector is composed of the sensitive value, d_i, of node i, and the *restricted* and *unrestricted camouflage values.* Restricted camouflage values in a message vector are required to be greater or lesser than the sensitive value for MIN and MAX aggregation respectively. Unrestricted camouflage values can be either greater or lesser than the sensitive value. The message vector is an array of values, where the sensitive data and the two types of camouflage values are assigned to specific positions in the array according to predefined policies. Let $I = \{1, 2, ..., n\}$ be the set representing the positions of V^i, $\forall i$, $1{\leq}i{\leq}N$. The *global secret set (GSS)*, a subset of I, denotes the secret index values kept at the base station to determine the final aggregated results. GSS contains the global secret information, which is partially shared among the network nodes. The *node's secret set* (*R^i*) is the secret information about GSS shared with node i. The base station specifies R^i for each i to include all elements from GSS and a subset of elements from $\overline{GSS}$, i.e., $\overline{GSS} \cap R^i \neq \emptyset$. P^i denotes the position of the sensitive value in V^i for node i. P^i is always a subset of GSS for all i, $1{\leq}i{\leq}N$, and $|P^i| = 1$. P^i is also a subset of R^i; hence R^i is the union of the index set of the restricted camouflage values and index of the sensitive value. U^i is the index set of the unrestricted camouflage values, where $U^i \subset \overline{GSS}$, and $U^i = \overline{R^i} = I - R^i$ ("$-$" denotes set difference).

This notation is formalized with the following four definitions which ensure the correctness and functionality of KIPDA. *Definition 1* ensures that the base station can correctly determine the final aggregated value. *Definition 2* enforces the requirement that any single node i cannot determine the entirety of GSS from P^i, by enforcing that $R^i - P^i$ draws from both sets GSS and $\overline{GSS}$. *Definition 3* guarantees that the true maximum value is not filtered out by the aggregation process. *Definition 4* enables the correctness of the previous 3 definitions by ensuring that the message vector is filled with the proper values.

Definition 1: The index, P^i, of the sensitive value, d_i, is drawn from GSS:

$$P^i \subset GSS,\ \forall i,\ 1{\leq}i{\leq}N,\ |P^i| = 1. \tag{4.1}$$

Definition 2: R^i contains elements from both GSS and $\overline{GSS}$. This is required to

Table 4.1: KIPDA notations.

message vector	Vector of camouflage and sensitive value sent to the next aggregator, indexed by I.
restricted camouflage values	Values in the message vector that are greater than the sensitive value for MIN aggregation and less for MAX aggregation.
unrestricted camouflage values	Values in a message vector that are either more or less than the sensitive value, d_i in the message vector.
V^i	Message vector of node i. $V^i = \{v_1^i, v_2^i, ..., v_n^i\}$.
v_ℓ^i	Values in V^i for node i where $\ell = 1, 2, ..., n$.
V^Ω	Last message vector received by the base station.
d_i	Sensitive value of node i. It is hidden in plain sight in V^i where $v_\ell^i = d_i$, if $\ell \in P^i$. The sensitive value can be a sensed value or an aggregate.
I	Index set of V^i. $I = \{1, 2, ..., n\}$
N	Number of nodes in the network.
n	Number of values in a message vector, $n = \|V^i\|$, $\forall i, 1 \leq i \leq N$. Determined according to privacy versus energy needs.
GSS	The global secret set kept at the base station that contains possible locations for the final network aggregated value.
R^i	The secret set for node i. Consists of the union of the indices of the restricted camouflage values and the index of the sensitive value.
U^i	Index set of unrestricted camouflage values values of node i. $U^i = \overline{R^i} = I - R^i$
P^i	Index of the sensitive value, d_i, of node i.

hide the sensitive value in V^i:

$$\overline{GSS} \cap (R^i - P^i) \neq \emptyset,\ \forall i,\ 1 \leq i \leq N. \tag{4.2}$$

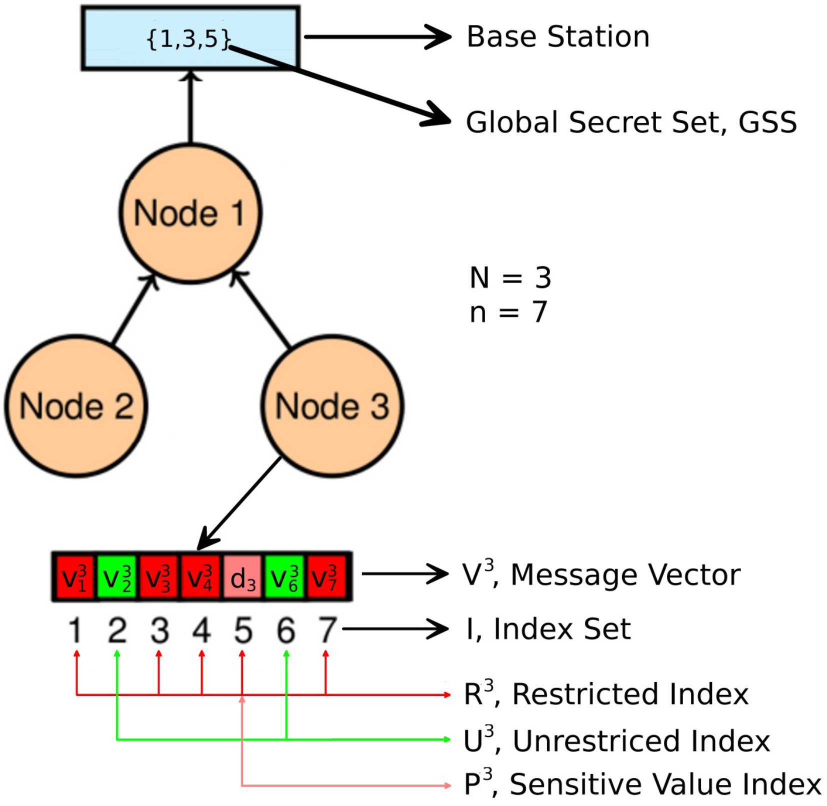


Figure 4.1: Illustrated KIPDA notations.

Definition 3 R^i is a proper superset of GSS:

$$GSS \subset R^i,\ \forall i,\ 1 \leq i \leq N. \tag{4.3}$$

Definition 4 A sensor node i fills in its message vector, $V^i = \{v^i_1, v^i_2, ..., v^i_n\}$, according to the following equation:

$$v^i_\ell = \begin{cases} d_i & \text{if } \ell \in P^i, \text{ and computing MAX or MIN} \\ \text{urand}(d_{min}, d_i) & \text{if } \ell \in R^i, \text{ and computing MAX} \\ \text{urand}(d_i, d_{max}) & \text{if } \ell \in R^i, \text{ and computing MIN} \\ \text{urand}(d_{min}, d_{max}) & \text{if } \ell \in U^i, \text{ and computing MAX or MIN.} \end{cases} \tag{4.4}$$

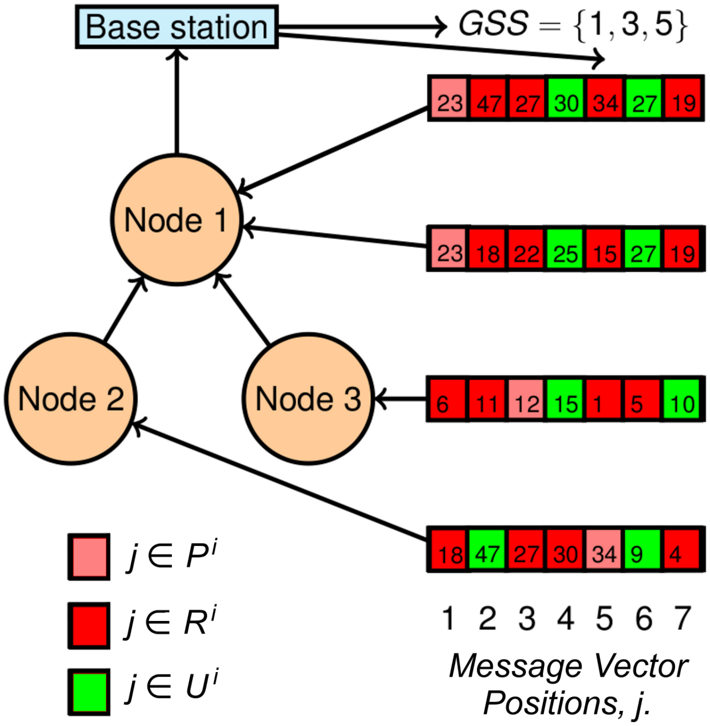


Figure 4.2: Example of a KIPDA aggregation scheme with three nodes.

where urand(x, y) generates a random number uniformly distributed between x and y, and d_{min} and d_{max} are the theoretical network minimum and maximum values respectively.

To illustrate KIPDA, consider the three node example of MAX aggregation shown in Figure 4.2. In this example, nodes 2 and 3 each want to send a single sensed sensitive value to node 1, which aggregates these values along with its own, and sends the aggregated sensitive value to the base station. Each node protects their sensitive value by writing it to the message vector with camouflage data. In this example, the message vector contains one sensitive value and six camouflage values.

Figure 4.2 illustrates the four phases of KIPDA for MAX aggregation: pre-distribution, sensing, aggregation, and base station processing. In the pre-distribution phase the base station determines, based on the above definitions, the elements of sets, GSS, and P^i and R^i for all nodes i where $1 \leq i \leq 3$. In the figure, GSS is determined to be $\{1, 3, 5\}$. P^i, for $i = 1$, 2, and 3, is determined to be {1}, {5}, and {3}, respectively. R^i, for $i = 1$, 2, and 3, is determined to be {1, 2, 3, 5, 7}, {1, 3, 4, 5, 7}, and {1, 2, 3, 5, 6}, respectively. U^i can be trivially determined from R^i for each i. The sets R^i are composed of three values from GSS, and two values from $\overline{GSS}$ each. After the sets are determined, the base station distributes P^i and R^i to each node i. During the sensing phase, node 2 places its sensitive value, 34, in the 5th slot in V^2. Then it determines the rest of V^2 according to Equation (4.4). These values could be picked randomly according to constraints such as the theoretical maximum and minimum values. In this way, $V^1 = \{23,\ 18,\ 22,\ 25,\ 15,\ 27,\ 19\}$, $V^2 = \{18,\ 47,\ 27,\ 30,\ 34,\ 9,\ 4\}$, and $V^3 = \{6,\ 11,\ 12,\ 15,\ 1,\ 5,\ 10\}$. During the data aggregation phase, when node 1 receives message vectors V^2 and V^3 from its children, it determines the aggregated value where $v^1_\ell = max\{v^i_\ell\}$ for each $\ell = 1, 2, ..., 7$ and $i = 1, 2, 3$. Hence, the aggregated message vector, V^1, is $\{23, 47, 27, 30, 34, 27, 19\}$, and replaces the original V^1, becoming the final message vector, V^Ω, that is sent to the base station. In the final phase, the base station determines the final network aggregate among the maximum elements indexed by GSS. In the example, elements at positions 1, 3, and 5 of the aggregated set V^Ω are 23, 27, and 34. Hence, 34 is the network MAX aggregate.

As described, this method might be prone to statistical analysis attacks. For example, an adversary could examine the packets for statistical correlations and use this information to guess R^i and U^i for certain i, ultimately guessing GSS and $\overline{GSS}$ of the base station. There are several methods to avoid this problem. In the network, if the theoretical maximum and minimum values are known, then the values in the message vector other than d_i could be chosen so the entire message resembles a uniform distribution. If the size of R^i prevents this, it can be increased. Alternatively, the sets could be changed or shuffled either after each network wide aggregation, or after a fixed number of aggregations. The base station would choose a new set GSS, and end-to-end encryption would distribute sets R_i and P_i to each node i. While this is a cryptographic approach, it would occur sparingly to conserve energy. These methods would also help if values of neighboring nodes are similar or correlated, or if an adversary manipulated the environment so that some sensor values were known, such as putting an ice block on top of a sensor that reports temperature.

4.3 Aggregation Accuracy and Collusion Attacks

This section explains the aggregation accuracy of KIPDA, the level of privacy protection, the robustness to node collusion (capture attacks), and the optimal sizes of sets GSS and R. First, accuracy is guaranteed according to the following conjecture:

Conjecture 1: KIPDA accurately computes the MAX and MIN aggregation functions.

Informal Sketch: The aggregation result can be affected only by the unrestricted camouflage values. However, the unrestricted camouflage values occur only in locations indexed by U^i. Since $GSS \subset R^i$, and $U^i \subset \overline{GSS}$, $\forall i, 1 \leq i \leq N$, the unrestricted camouflage values do not affect the aggregated results in positions indexed by GSS at each node. This is because the elements of GSS and $\overline{GSS}$ are disjoint. Assuming paths to the base station follow a tree route with the base station as the root, any subtree will contain the maximum or minimum aggregate value of that subtree in GSS. The base station can always determine the sensitive network wide aggregate from GSS of the final message vector.

Let us continue with some definitions:

Definition 5: A *rogue node* is a node compromised by an adversary that will collect sensitive data, yet will still follow the network protocols, so to appear as uncompromised.

Definition 6: Given k items where $k{\geq}1$, the items are said to be *k-indistinguishable* if a sensitive value, d, from among the k items cannot be determined more accurately than by guessing from the other $k{-}1$ items. A set with a single item has a k value of one.

The following 2 claims provide support for Conjecture 2, which quantifies k when a single rogue nodes tries to learn sensitive data. First, I present a change in notion. Since the sizes of R^m and U^m are the same for all m, $1{\leq}m{\leq}N$, they will be denoted as $|R|$ and $|U|$ with the indices removed.

Claim 1: A victim node i has k-indistinguishability where $1{\leq}k{\leq}|U|{+}1$ when an adversary has captured node j and uses set U^j to determine d_i.

Informal Sketch: Only the unrestricted values can be higher or lower than d_i for MAX and MIN aggregation respectively. The maximum number of these unrestricted values is $|U|$. In the best case for victim node i, node j cannot determine what sets the positions of these $|U|$ values of V^i are in: R^i, U^i, or P^i. In addition to the $|U|$ largest or smallest values, the rogue node cannot distinguish the position of the sensitive value itself from the other $|U|$ values, hence k is increased by one.

Take for example MAX aggregation in Figure 4.3, rogue node j (Node 2 from Figure 4.2) knows the sensitive value of victim node i (Node 3 from Figure 4.2) will be in one of the $(|U| + 1)$ largest values in node i's message vector, V^i. These values correspond to indices 2, 3, and 4. Here, indistinguishability is 3.

Claim 2: A victim node i has k-indistinguishability where $1{\leq}k{\leq}|R|{-}1$ when an adversary has captured node j and uses set R^j to determine d_i.

Informal Sketch: An adversary, after capturing node j, knows that the sensitive data, d_i of node i is guaranteed to be in one of the positions denoted by R^j in V^i. This is

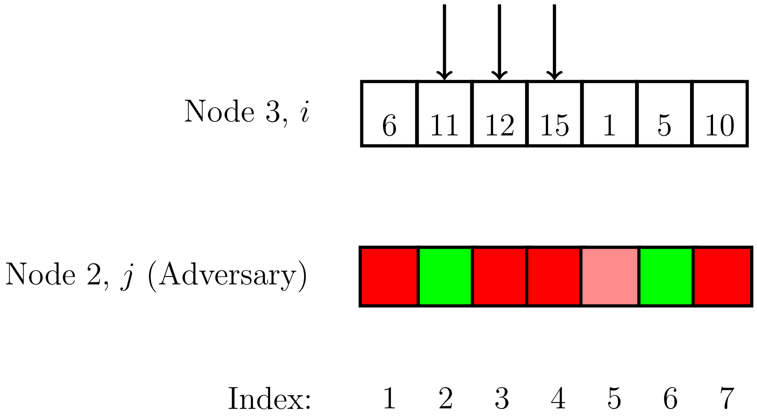


Figure 4.3: Nodes 2 and 3 are from Figure 4.2. An adversary can determine that the sensitive value is in the $|U| + 1$ largest positions of V^i for MAX aggregation.

because $GSS \subset R^m$ for all m, $1 \leq m \leq N$. The adversary cannot determine the actual location of d_i since she does not know whether any of the corresponding elements in her R^j set are in the victim's R^i, P^i, or U^i sets. Since the adversary knows at least one position of GSS, k is reduced by one.

For example in Figure 4.4, the adversary after capturing node j (Node 2 from Figure 4.2) knows the sensitive value is contained in one of the positions of node i's message vector denoted by the rogue node's R^j set. Since the smallest value of the positions denoted by R^j is contained in P^j, k is reduced by one.

Conjecture 2: For any single rogue node j trying to collect the index of the sensitive information, d_i, from node i, KIPDA provides the following k-indistinguishability level where k is given as:

$$1 \leq k \leq min(|U| + 1, |R| - 1). \tag{4.5}$$

Informal Sketch: Since an adversary can choose either or both techniques from Claims 1 and 2, it follows that k will be less than or equal to the minimum of the two. Future work will consider if these two attacks are exhaustive.

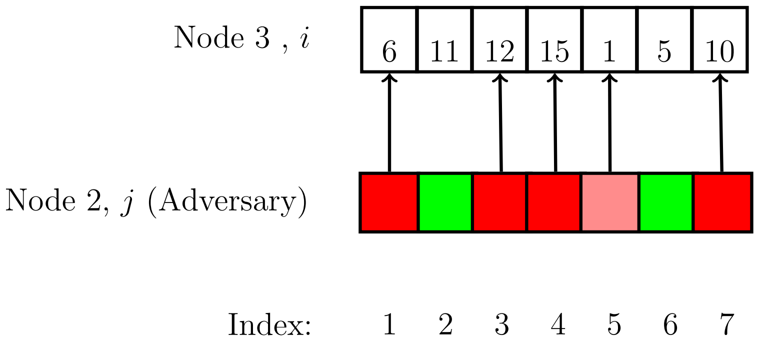


Figure 4.4: Nodes 2 and 3 are from Figure 4.2. An adversary can determine that the sensitive value is in R^j of V^i.

The next two claims help determine the average value k a victim node has against a single colluding node.

Claim 3: A victim node i has average indistinguishability k where $k = \Lambda$, against a rogue node j when U^j is used to determine d_i. Λ is given as:

$$\Lambda = (|U| - \Psi + 1)$$

where Ψ is the expected number of elements of the largest or smallest $(|U| + 1)$ values (for MAX or MIN aggregation respectively) in V^i that are in positions that are members of the set U^j. When Λ equals $|U| + 1$, it is a special case of Claim 3 where no elements are shared between the positions of largest or smallest $(|U| + 1)$ values in V^i and set U^j. Ψ is defined as:

$$\Psi = \frac{|U| \cdot (|U| + 1)}{|I|}. \tag{4.6}$$

Informal Sketch: k is reduced from $|U|$+1 by one for every element shared between U^j and the positions denoted by the largest or smallest $|U|+1$ values in V^i. Because $U^m \notin GSS$ for all m, $1 \leq m \leq N$, these value can be discounted as they do not contain any sensitive information. The expected value, Ψ, is the product of the $(|U| + 1)$

number of largest or smallest elements, and all the possible positions of set U, divided by all possible positions that the largest or smallest values can fall in, $|I|$.

For example in Figure 4.3 for MAX aggregation, position 2 can be ruled out because it is a member of U^j. Hence, k is reduced to 2. Also, it does not matter if any of the $|U|+1$ largest values are in position P^j, because any of these values could fall in P^i, and the adversary does not know if $P^i = P^j$.

Claim 4: A victim node i has average indistinguishability k, where $k = \Pi$, against a rogue node j when R^j is used to determine d_i. Π is given as:

$$\Pi = \sum_{g=1}^{|R|} P(g) \cdot (g-1) \tag{4.7}$$

where $P(g)$ is the probability that the g^{th} largest item appears in the position denoted by P^j in V^i. This probability can be actualized as:

$$P(g) = \frac{1}{|R|}, \tag{4.8}$$

Informal Sketch: For MAX aggregation, the g^{th} largest value in V^i that is in the position denoted by P^j gives a k value of $g-1$. Because $P^j \subset GSS$, any smaller values in the positions denoted by R^j can be ruled out. This is because their positions are either in GSS or $\overline{GSS}$. If their positions are in GSS, none of them are the sensitive value because they are smaller, and, if they are in $\overline{GSS}$, they are camouflage data. This leaves values larger than the g^{th} value, ($g-1$ in total), as possible sensitive values. d_i cannot be distinguished from the other $g-1$ largest values, because the adversary does not known whether these values are in the R^i, P^i, or U^i positions. MIN aggregation would follow a similar logic and examine the smallest g^{th} values.

For example in Figure 4.4, k is given as 4, because position P^j in V^i contains the 5th largest value.

Conjecture 3: Any victim node i has the following average k value against a single rogue node j:

$$k = min(\Lambda, \Pi), \tag{4.9}$$

where Λ and Π are from Claims 3 and 4.

Informal Sketch: Since an adversary can choose either or both techniques from Claims 3 or 4, it follows that k will be equal to the minimum of the two. Future work will consider if these attacks are exhaustive.

KIPDA provides greater protection from eavesdroppers. To an outside observer without knowledge of GSS, R^i, and P^i for any i, k is equal to $|V|$. The following claims and conjectures quantify k in the case when several nodes collude and their objective is to determine the base station's GSS and $\overline{GSS}$ sets. This is accomplished in two ways. The first is to infer GSS from P^j, and the second is to infer $\overline{GSS}$ from U^j.

Claim 5: The expected number of colluding nodes, $E[x]$, that can determine all $|GSS|$ elements in GSS from P^j, assuming P^j is randomly selected from GSS is given as:

$$E[x] = |GSS| \cdot H_{|GSS|} = |GSS| \cdot \sum_{i=1}^{|GSS|} \frac{1}{i}, \tag{4.10}$$

where $H_{|GSS|}$ is the harmonic number of $|GSS|$.

Informal Sketch: This is an instance of the coupon collector's problem [120]. Each time a node colludes with another node, it is similar to collecting the coupon, P^j, out of $|GSS|$ coupons with replacement.

Claim 6: The expected number of nodes colluding, $E[x]$, that can determine all $|\overline{GSS}|$ elements from U^j, assuming no bias when U^j is selected from $\overline{GSS}$, is given as:

$$E[x] = \frac{|\overline{GSS}|}{|U|} \cdot H_{|\overline{GSS}|} = \frac{|\overline{GSS}|}{|U|} \cdot \sum_{i=1}^{|\overline{GSS}|} \frac{1}{i}. \tag{4.11}$$

Informal Sketch: This is a slight variant of the coupon collectors problem. Instead of collecting one coupon at each collusion, $|U|$ unique coupons are collected, with replacement. Equation (4.11) is based on the probability, p_m, of choosing the m^{th} element of $\overline{GSS}$, given as:

$$p_m = \frac{|U|(|\overline{GSS}| - m + 1)}{|\overline{GSS}|}. \tag{4.12}$$

The optimal size of GSS can be determined using Claims 5 and 6. According to Claim 5, with a fixed message vector size, $|V|$, the expected number of colluding nodes needed to obtain GSS increases when the number of elements in GSS increases. According to Claim 6, the expected number of colluding nodes required to discover GSS decreases when the number of elements in $\overline{GSS}$ increases. Thus, the least expected number of collusive nodes needed to infer either GSS or $\overline{GSS}$ is minimized by taking the intersection of the lines specified by Equations (4.10) and (4.11).

For example in Figure 4.5, $|I|$ is set to 20 and $|R|$ varies between 18 and 14, where each variation is given an approximate average value of k against a single rogue node. Several values of $|R|$ are included to illustrate the trade-offs between the level of k-indistinguishability a victim node has against a single rouge node, versus the expected number of colluding nodes needed to determine the sensitive information. Better protection against a single rogue node will have less protection against several rogue nodes colluding, and vice versa. Network designers can use this trade-off to choose the optimal value of $|R|$. Figure 4.5 shows that if $|R| = 17$, the optimal value of $|GSS|$ is 6, and it is expected that 15 nodes would need to collude before GSS or $\overline{GSS}$ are entirely known.

The optimal size of all sets can now be determined. Since $|I|$ (or $|V|$) affects bandwidth and power consumption, it cannot be too large. The optimal size is decided according to the energy budget of the system balanced against the level of desired indistinguishability and is discussed in Section 5.1 of this chapter. Once $|I|$

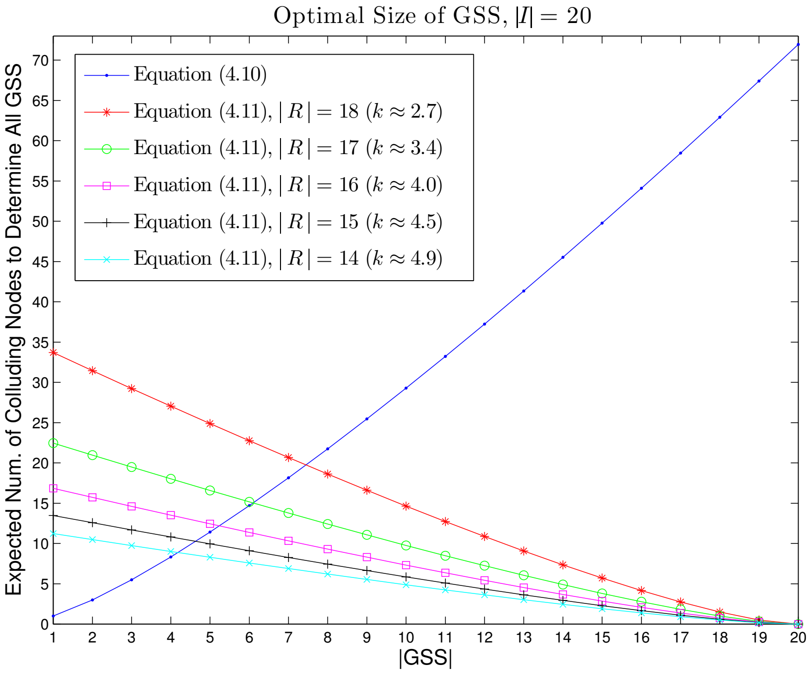


Figure 4.5: The optimal size of GSS for $|I| = 20$ is given by the intersection of the curve of Equation (4.10), with various curves of Equation (4.11). A trade-off exists between lower values for $|R|$, which give a higher indistinguishability value against a single rogue node (given as k in legend), and higher values for $|R|$, which require more expected nodes to collude to determine GSS.

is determined, the sizes of sets GSS and R^i must be chosen carefully as discussed above to achieve good performance against node collusion attacks.

The following gives an argument for the average k value when x nodes collude.

Claim 7: A victim node i has average indistinguishability k, where k=Λ, against x colluding nodes when U^j for all j colluding nodes is used to determine $\overline{GSS}$. Λ is given as:

$$\Lambda = (|U| - \Psi' + 1)\,, \tag{4.13}$$

where Ψ' is the expected number of elements of the largest or smallest $(|U|+1)$ values (for MAX or MIN aggregation respectively) in V^i that are in positions denoted by the union of the sets U^j for all colluding nodes j. Ψ' is defined as:

$$\Psi' = \frac{|\overline{GSS}_{Known}| \cdot (|U| + 1)}{|V|}, \tag{4.14}$$

where $|\overline{GSS}_{Known}|$ and $|GSS_{known}|$ (seen later) are the inverses of Equations (4.10) and (4.11) from Claims 5 and 6, and are given as:

$$\begin{aligned} |GSS_{known}| &= \frac{x - \frac{1}{2}}{W(\frac{1}{2}e^{\gamma}(2x-1))} \\ |\overline{GSS}_{known}| &= \frac{|U| \cdot x - \frac{1}{2}}{W(\frac{1}{2}e^{\gamma}(2 \cdot |U| \cdot x - 1))}, \end{aligned} \tag{4.15}$$

where γ is the Euler-Mascheroni constant, 0.577215, and W is the Lambert W-Function [37] or product log.

Informal Sketch: Because $U^m \subset \overline{GSS}$ for all m, $1 {\leq} m {\leq} N$, the rogue nodes combine their U^j sets together (by taking the union of the sets) to build $\overline{GSS}_{Known}$. Building off of Claim 3, the expected value of the shared number of elements between the largest or smallest $|U|+1$ elements in V^i, for MAX or MIN aggregation respectively, whose positions are in U^j is modified, where an instance of $|U|$ is replaced with $|\overline{GSS}_{Known}|$.

Claim 8: A victim node i has average indistinguishability k, where $k{=}\Pi$, against x colluding nodes when R^j, for all j colluding nodes, is used is to determine GSS. Π is given as:

$$\Pi = \sum_{g=1}^{|V| - |\overline{GSS}_{Known}|} P(g) \cdot (g - 1) \tag{4.16}$$

where $P(g)$ is the probability that the g^{th} largest values (or smallest values for MIN aggregation) in V^i fall into any positions denoted by GSS_{known}. It can be actualized as:

$$P(g) = \frac{|GSS_{Known}|}{|V| - |\overline{GSS}_{known}|}. \tag{4.17}$$

Informal Sketch: Given Claim 4, $|R|$ is replaced with $|V| - |\overline{GSS}_{Known}|$. This is because when rogue nodes collude, the number of available restricted camouflage values in V^i (given the information known to the rogue nodes) is $|V| - |\overline{GSS}_{Known}|$. The colluding nodes cannot determine if the values in $V - \overline{GSS}_{Known}$ are in GSS or not. As $|V| - |\overline{GSS}_{Known}|$ decreases because more elements are found in $\overline{GSS}$, more elements in the union of R^j for all colluding nodes j are determined to be in $\overline{GSS}$. This reduces the possible elements in V^i that can be indistinguishable. Eventually, all elements that are left in R_j, as j increases, will be members of GSS.

Conjecture 4: Any victim node i has the following average indistinguishability value k when x nodes collude:

$$k = \begin{cases} 1 & \text{if } |GSS_{known}| = |GSS| \text{ or } |\overline{GSS}_{Known}| = |\overline{GSS}| \\ min(\Lambda, \Pi) & \text{otherwise.} \end{cases} \tag{4.18}$$

Informal Sketch: Since an adversary can choose either or both techniques from Claims 7 or 8, it follows that k will be equal to the minimum of the two. If all members of sets GSS or $\overline{GSS}$ are determined, k reduces to one.

With just one rogue node, Equation (4.18) reduces to Equation (4.9). This is because $|\overline{GSS}_{known}|$ would equal $|U|$, and $|V| - |\overline{GSS}_{known}|$ would equal $|R|$. An example of Conjecture 4 is illustrated in Figure 4.6. The sizes of GSS and R are chosen from the optimal values from Figure 4.5. When $|R| = 15$, $|GSS| = 4$, and 3 nodes collude, a victim node has a k-indistinguishability value on average of approximately 4.

There is one limitation of KIPDA. As explained before, the level of indistinguishability is based on the number of elements in U, and an adversary could examine the m largest or smallest values in a message vector. However, over several iterations of KIPDA (different aggregation epochs), an adversary might watch these positions. She could eliminate the positions where the m values are not in the positions of the

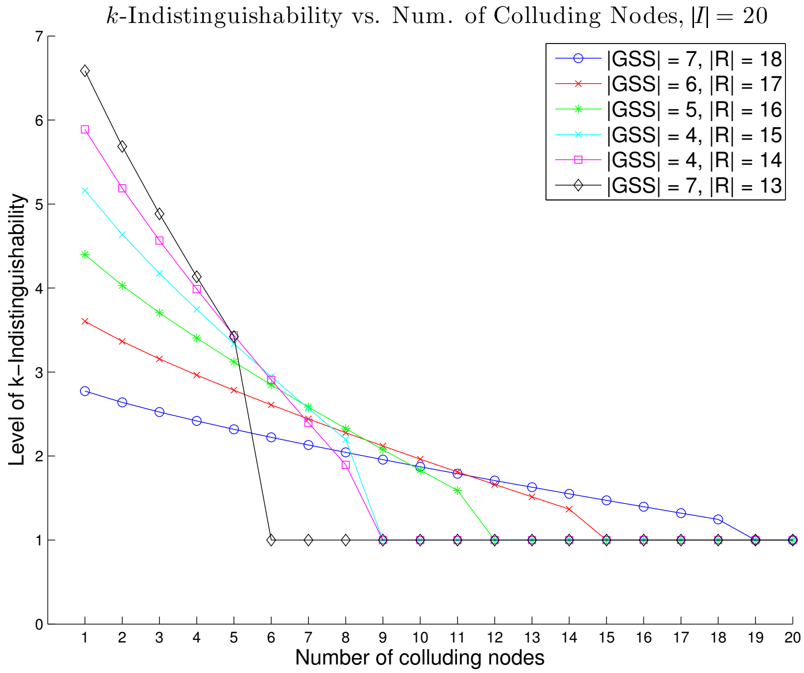


Figure 4.6: Level of indistinguishability, k, against the number of colluding nodes, x. $|I| = 20$ and varies sizes of GSS and R are plotted.

previous iteration's m values. In this way, the adversary could eventually determine the location of the sensitive value for any node.

This threat can be addressed by changing the sets GSS, P^i, and R^i at every aggregation epoch, similar to a one time pad. The base station could distribute the sets P^i and R^i to each node, although this might be expensive. I suggest an alternative solution that uses special random number generators with unique properties. Unique seeds for these generators would be shared by the specific nodes and the base station. They could be distributed with standard encryption techniques or with public/private key encryption such as TinyECC [106], since this would occur occasionally. The unique properties of the generators would allow nodes to generate one number from the set GSS and enough numbers from $\overline{GSS}$ to determine the elements for the set U^i $\forall i, 1{\leq}i{\leq}N$. However, a node would not be able to determine all of GSS, but have enough knowledge to be able to draw the sets it needs: R, and P. The exact implementation of these special random number generators is left for future work.

4.4 Protocol

In this section I describe the protocols for KIPDA. I assume aggregation trees are constructed according to standard data aggregation protocols [109]. There are four phases to KIPDA: pre-distribution, sensing, aggregating, and base station processing.

4.4.1 Pre-distribution

In the pre-distribution phase, the base station:

1. Determines the sizes of sets, I, GSS, and R, where $|GSS| < |R| < |I|$. The optimal sizes of these sets were discussed previously.
2. Chooses unique elements of GSS, where $GSS \subset I$. These elements can be chosen uniformly at random.
3. Determines P^i for each node i, where $|P^i| = 1$, and $P^i \subset GSS$.
4. Determines R^i for each node i, by taking all the elements from GSS and $(|R|{-}|GSS|)$ elements from $\overline{GSS}$. The elements in R^i are unique and can be chosen uniformly at random.
5. Distributes the sets (1) P^i and (2) either R^i or U^i to each node i, depending on which set contains less bits. There are many methods for distributing keys securely in WSNs, (e.g. Perrig et al. [123] use a Faraday cage to ensure key

Algorithm 4.1 KIPDA: Pre-distribution Phase for Base Station

```
procedure Pre-distribution
    Determine $|I|$, $|GSS|$, and $|R|$, where $|GSS| < |R| < |I|$.
    Determine elements of $GSS$ such that $GSS \subset I$.
    Determine for each node $i$, $P^i$, where $|P^i| = 1$, and $P^i \subset GSS$.
    Determine for each node $i$, $R^i$, containing all elements from $GSS$ and
        $(|R| - |GSS|)$ random elements from $\overline{GSS}$.
    Distribute sets (1) $P^i$ and (2) either $R^i$ or $U^i$ dependent on which
        contains fewer bits, to all nodes $i$ .
end procedure
```

secrecy, key authenticity, forward secrecy, and demonstrative identification), which could easily be modified to distribute these sets.

This phase is summarized in Algorithm 4.1.

4.4.2 Sensing

In the sensing phase, each node i determines the values for the set V^i where $V^i = \{v_1^i, v_2^i, ..., v_n^i\}$, and $n = |I|$. This step applies only to sensor nodes and not to internal nodes that only aggregate and forward packets. R^i denotes the positions in V^i that behave according to the rules for restricted values, with the exception of P^i ($P^i \subset R^i$), which denotes the position in V^i that is assigned d_i. If sensed values are in the range [d_{min}, d_{max}], the restricted camouflage values are drawn from [d_{min}, d_i] for MAX aggregation, and from $[d_i, d_{max}]$ for MIN aggregation. The unrestricted camouflages values are drawn from [d_{min}, d_{max}], as described by Equation (4.4). Different nodes can use different distributions to generate random values for the restricted or unrestricted camouflages values so it is harder for others to infer the sensitive value from V^i. This phase is shown in Algorithm 4.2.

Algorithm 4.2 KIPDA: Sensing Phase

```
for each node $i$ do
    procedure Sensing
        Determine each $v_\ell^i$ in $V^i = \{v_1, v_2, ..., v_n\}$ according to Equation (4.4).
    end procedure
end for
```

4.4.3 Aggregation

In the aggregation phase, each node i receives from its children information to be aggregated (unless it is a leaf node), aggregates the information, and passes the aggregate to the next hop in the routing protocols either immediately, when queried, or according to other protocols. If node i is a leaf node, aggregation is skipped, and V^i is transmitted according to the forwarding protocols (e.g. after waiting a certain amount of time upon receiving the aggregation query). If node i has its own message vector from sensing, that information is also aggregated along with the children information. Aggregation is accomplished by computing the MAX or MIN of v_ℓ^j for each $\ell = \{1, 2, ..., n\}$ among all children nodes j. In this way, V^i is replaced with the aggregated information. When finished or after a certain amount of time, V^i is forwarded to the next hop in the aggregation routing tree protocol. This phase is shown in Algorithm 4.3.

For the MAX (or MIN) aggregation functions, the values in the message vectors grow larger (or smaller) as they approach the base station, possibly compromising privacy. One solution is to replace one or more values indexed by U^i so that they appear more uniformly distributed through $[d_{min}, d_{max}]$. Nodes closer to the base station might want to give the appearance of a distribution shifted to the upper or lower ends of the maximum or minimum theoretical network value.

Algorithm 4.3 KIPDA: Aggregation Phase

```
for each node $i$ after receiving an aggregation query do
    procedure AGGREGATION
        if node $i$ has children then
            Receive from each child $j$, $V^j$,
        else
            Forward $V^i$ to the next hop according to forwarding protocols.
            return
        end if
        for each $\ell$ in $v^j_\ell$ where $j \in$ children of node $i$ do
            if node $i$ also has sensed information then
                $i \in j$.
            end if
            Determine aggregate (MAX or MIN) of $v^j_\ell$, $\forall j$.
            Place aggregated information in position $\ell$ in $V^i$.
        end for
        Forward $V^i$ to the next hop according to forwarding protocols.
    end procedure
end for
```

4.4.4 Base Station Processing

The final phase is performed on the base station which receives the last aggregated message vector, V^Ω. If other independent messages arrive at the base station, it first performs Algorithm 4.3 on these messages to obtain V^Ω. The final network aggregate (the maximum or minimum value sensed in the network) is computed by selecting the MAX or MIN from the values indexed by GSS in V^Ω:

$$\begin{aligned} &max_{i\in GSS}(v_i^\Omega) \quad \text{for MAX aggregation,} \\ &min_{i\in GSS}(v_i^\Omega) \quad \text{for MIN aggregation.} \end{aligned} \tag{4.19}$$

Algorithm 4.4 KIPDA: Base Station Processing Phase

```
procedure BASE STATION PROCESSING
    if more than one message received then
        Perform Algorithm 4.3 on the messages to obtain $V^{\Omega}$.
    else
        Receive $V^{\Omega}$
    end if
    Determines the network aggregate:
        $max_{i \in GSS}(v_i^{\Omega})$ for MAX aggregation.
        $min_{i \in GSS}(v_i^{\Omega})$ for MIN aggregation.
end procedure
```

This phase is summarized in Algorithm 4.4.

4.5 Evaluation Analysis

In this section, I compare KIPDA's energy use and time delay to that of end-to-end data collection and hop-by-hop encryption aggregation.

4.5.1 Energy Analysis

End-to-End Data Reporting

End-to-end data reporting without aggregation is energy-intensive because every reported value is transmitted to the base station. In the following analysis, values are assumed to be 16 bits wide. (In cases with block encryption, more data are actually sent over the radio because block sizes are larger than the size of the plaintext.) Table 4.2 shows the number of bits transmitted (and the corresponding energy consumption) per node for each level in an aggregation routing tree with a branching factor of 5. This scheme is similar to that shown in Figure 2.2, left. The network level is the number of hops away from the base station. Energy consumption per transmitted bit is determined using calculations from Meulenaer et al. that assume a MicaZ node architecture [42].

Nodes closer to the base station consume more bandwidth because more data pass through them. To balance traffic loads among nodes, either the sink node sometimes moves around or the nodes themselves migrate [26], both of which are impractical in many cases. Therefore I assume a fixed network topology.

Average bandwidth consumption in end-to-end data collection with a branching factor of 5 is $O(\log N)$ per node, assuming no aggregation and N nodes in the network. KIPDA's average bandwidth consumption is $O(|V^i|)$ per node, i.e., the number of values in a message vector, and is independent of the number of nodes. For this scenario, energy usage grows more quickly with network size than with KIPDA. Additionally, because nodes near the sink (base station) have to send more information, there will be larger delays due to the time to transmit extra bits over the radio.

Despite the energy cost, end-to-end encryption provides the best privacy protection. Outsiders and neighboring nodes are prevented from determining the sensitive values. However, because aggregation is not performed, this scenario costs extra energy.

Hop-by-Hop Encryption Aggregation

Hop-by-hop aggregation with encryption consumes less power near the sink than the previous scenario, although, a large amount of power is consumed throughout the network by the repeated decryption and re-encryption phases, which also introduces

Table 4.2: Bandwidth energy usage of end-to-end data collection. Collection follows a tree route with a branching factor of 5. Level is the number of hops away from the base station.

Level	Number of Nodes	Bits Sent Per Node	MicaZ Radio Energy Use per Node (μJ)
1	5	312,496	187,496.6
2	25	62,496	37,497.6
3	125	12,496	7,497.6
4	625	2,496	1,497.6
5	3,125	496	297.6
6	15,625	96	57.6
7	78,125	16	9.6

time delay. KIPDA is next compared to three hop-by-hop aggregation schemes that use IDEA [100], RC5 [132], and RC4 [121] encryption. IDEA (International Data Encryption Algorithm) is a symmetric encryption technique that uses 64 bit blocks and a 128 bit key. RC5 uses variable blocks and key sizes, but my analysis considers only blocks of 64 bits. RC4 is the most widely used stream cipher technique and is found in such protocols as the Secure Sockets Layer (SSL) on the Internet, and WEP that secures wireless networks. It operates on segments of 8 bits.

The first step is to determine the energy consumption of the three encryption ciphers on a generic sensor architecture. I use the results from Ganesan et al. [70] which generalize the costs of IDEA, RC4, and RC5 to any mote architecture and is given as:

$$Time_{ENC/DEC} = \frac{a + b \cdot \lceil \frac{\text{text length}}{\text{block size}} \rceil}{\text{processor frequency} \cdot \text{bus width}}, \tag{4.20}$$

where variables a and b are given as follows:

$$\begin{aligned} a &= a_{BASE} + a_{MUL} + a_{RISC} \\ b &= b_{BASE} + b_{MUL} + b_{RISC}, \end{aligned} \tag{4.21}$$

Table 4.3: Parameters a_{BASE} and b_{BASE} taken from Ganesan et al. [70].

Algorithm	a_{BASE}	b_{BASE}	blocksize (bits)
RC5 init/encrypt	352114	40061	64
RC5 init/decrypt	352114	39981	64
IDEA encrypt	67751	80617	64
IDEA decrypt	385562	84066	64
RC4	68540	13591	8

Table 4.4: Parameters a_{MUL} and b_{MUL} taken from Ganesan et al. [70].

Operation	a_{MUL}	b_{MUL}
w/ MUL instruction	19016	-1143
w/o MUL instruction	-14330	8252

Table 4.5: Parameters a_{RISC} and b_{RISC} taken from Ganesan et al. [70].

	a_{RISC}	b_{RISC}
RISC	3207	1661
CISC	77175	-103593

and where parameters a_{BASE} and b_{BASE} are given in Table 4.3. Parameters a_{MUL} and b_{MUL} depend on whether a multiplication instruction is native to the architecture and are given in Table 4.4, and a_{RISC} and b_{RISC} depend on whether a RISC or CISC architecture is used, and are given in Table 4.5. a_{BASE}, b_{BASE}, a_{MUL}, b_{MUL}, a_{RISC}, and b_{RISC} were determined in Ganesan et al. [70] by minimizing the least square relative error in their experiments.

I then apply these generalizations to two common architectures, MicaZ and TelosB. The MicaZ architecture has a bus width of 8 bits and a processor that runs at 7.37 MHz, and the TelosB architecture has a bus width of 16 bits and a processor that runs at 4 MHz. The cost of their three frequent operations — compute one clock tick, and transmit and receive one bit — were taken from Meulenaer et al. [42] and are given in Table 4.6.

Table 4.6: Energy consumption in microjoules, μJ, and nanojoules, nJ, of common operations on the MicaZ (7.37 MHz) and TelosB (4 MHz) motes [42].

Operation	MicaZ	TelosB
Compute for 1 Clock Tick	3.5 nJ	1.2 nJ
Transmit 1 bit	0.60 μJ	0.72 μJ
Receive 1 bit	0.67 μJ	0.81 μJ

Table 4.7: Time in seconds, number of processor clock ticks, and energy in microjoules to encrypt (Enc) and decrypt (Dec) data on the MicaZ and TelosB architectures.

Method, Architecture	Time μs	Clock Ticks	Energy μJ
IDEA Enc, MicaZ	2902.12	21388.63	74.86
IDEA Enc, TelosB	2673.58	10694.31	12.83
IDEA Dec, MicaZ	8350.80	61546.13	215.41
IDEA Dec, TelosB	7693,27	30773.06	36.93
RC5 Enc, MicaZ	7037.25	51864.50	181.53
RC5 Enc, TelosB	6483.06	25932.25	31.12
RC5 Dec, MicaZ	7035.89	51854.50	181.49
RC5 Dec, TelosB	6481.81	25927.25	31.11
RC4, Enc & Dec, MicaZ	2018.00	14872.63	52.05
RC4, Enc & Dec, TelosB	1859.08	7436.31	8.92

With this information, the time in seconds, the number processor clock ticks, and the energy spent in Joules to perform the encrypt and decrypt primitives can be estimated for the IDEA, RC5, and RC4 ciphers on the MicaZ and TelosB architectures. The time in microseconds is determined from Equation (4.20). The number of clock ticks is determined by multiplying the time by clock frequency. Energy usage is determined from the number of clock ticks according to energy spent per tick, given in Table 4.6. The compiled results are given in Table 4.7. Since the time to encrypt and decrypt for RC4 is the same, it is given only once per architecture.

To determine the energy consumption of a node in a hop-by-hop aggregation

Table 4.8: Energy consumption of hop-by-hop encryption per node for 10 bits of data for the MicaZ and TelosB architectures, assuming 5 children nodes.

Method, Architecture	Energy μJ
IDEA, MicaZ	1404.74
IDEA, TelosB	502.76
RC5, MicaZ	1341.80
RC5, TelosB	491.97
RC4, MicaZ	375.55
RC4, TelosB	129.87

method, I combine the information in Table 4.7 with the energy costs of the aggregation process: receiving packets, decrypting packets, aggregating the information, encrypting the aggregate, and transmitting the encrypted aggregate. This is formalized in the following equation:

$$E_{HBH} = c \cdot (Rx(b) + Dec(b) + Agg) + Enc(b) + Tx(b), \tag{4.22}$$

where c is the branching factor (number of children at each node in the network), $Rx(b)$ and $Tx(b)$ are the energy costs of receiving and transmitting b bits, $Dec(b)$ and $Enc(b)$ are the energy costs of decrypting and encrypting b bits, Agg is the energy required to compute the aggregate, assuming the time to aggregate c values is c clock ticks, and b is the number of bits in a block or segment size. The energy to transmit and receive 1 bit is taken from Table 4.6. The energy costs to encrypt and decrypt are given in Table 4.7. The number of bits, b is determined from Table 4.3. The results of the energy costs of hop-by-hop aggregation are shown in Table 4.8, which gives the energy used when the branching factor, c, is set to 5.

The energy consumption of KIPDA consists of receiving data, aggregating it, and transmitting the aggregate. However, this depends on the number of values in the message vector, and the number of bits per value. The following equation formalizes the energy consumption per node:

$$E_{KIPDA} = m \cdot (c \cdot (Rx(b') + Agg) + Tx(b')) \,, \tag{4.23}$$

where m is the number of values in a message vector, and b' is the number of bits per value. The following section assumes $b' = 10$ and gives the results of KIPDA and compares it with hop-by-hop aggregation. 10 bits were chosen because it allows 1,024 distinct values, which is enough to express a sensor reading in many WSN applications.

Discussion of Results and Size of Set I

Figure 4.7 shows that KIPDA can use 34 to 35 values in a message vector before it consumes the same energy as IDEA and RC5 for the MicaZ architecture. However, it can send about 8 camouflage values before it uses the same amount of energy as RC4. Figure 4.8 gives results for the TelosB architecture where RC4 works so efficiently that the crossover point is about 2 values. For RC5 and IDEA on the TelosB architecture the crossover point is 9 decoy messages. In the next Chapter, these results are tested and compared to an energy aware wireless sensor network simulator, PowerTOSSIM-Z.

The optimal size of I can now be determined. For KIPDA to achieve a net power savings than using IDEA or RC5 hop-by-hop encryption, $|I|$ for the MicaZ motes needs to be less than 34. For the TelosB architecture, $|I|$ needs to be less than 10 for IDEA and RC5. KIPDA does not offer an energy advantage on the TelosB architecture if RC4 hop-by-hop encryption were used, although, it will protect information from in-network nodes. Also, as described in the next section, KIPDA significantly reduces delay in the network. This would be appealing in networks that require a fast response time, reasonable energy consumption, and privacy protection.

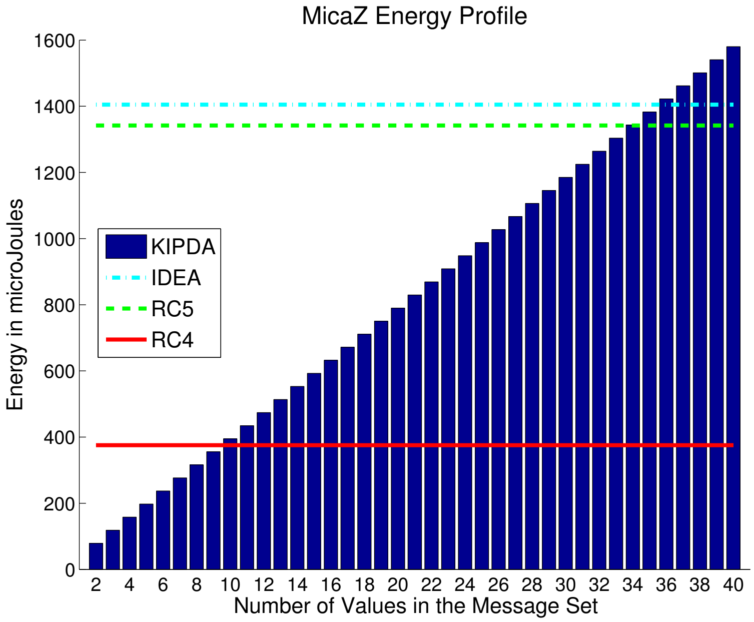


Figure 4.7: Energy profile of the MicaZ sensor architecture for a node performing data aggregation in a WSN. Values in a message vector are 10 bits, and the aggregation tree has a branching factor of 5. KIPDA can use 33 decoy values before it uses more energy than hop-by-hop aggregation with either IDEA or RC5 encryption.

The plaintext size in all cases was 10 bits. Because IDEA and RC5's block sizes are both 64 bits, it required an extra 54 bits to transmit a message. RC4, however, needed only two stream segments, to transmit 16 bits. This could explain why RC4 is more energy efficient, because extra bandwidth was not wasted on larger block sizes. This is discussed in more detail in Chapter 5.

4.5.2 Delay Analysis

The time it takes for a node to perform hop-by-hop aggregation with encryption is determined by the following equation:

$$T_{HBH} = c \cdot Dec_T(b) + Enc_T(b) + (c+1) \cdot \frac{b}{BW}, \tag{4.24}$$

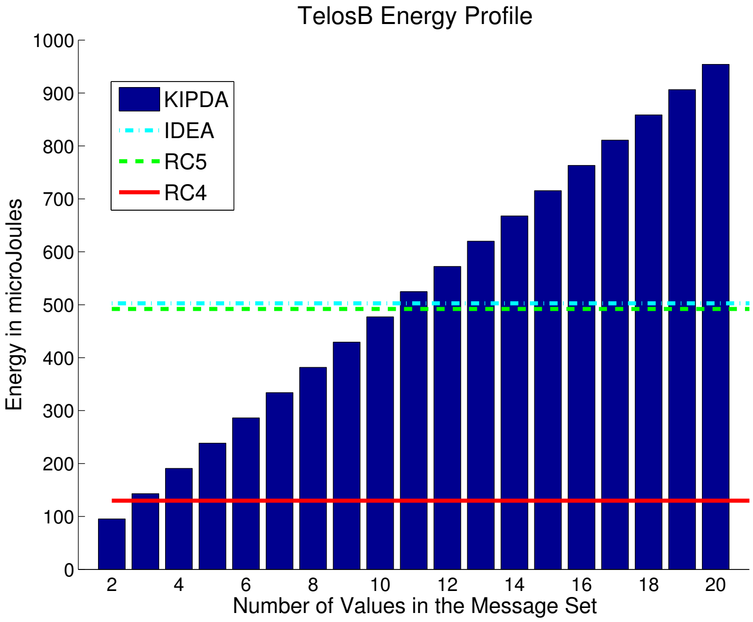


Figure 4.8: Energy profile of the TelosB sensor architecture for a node performing data aggregation in a WSN. Values in a message vector are 10 bits, and the aggregation tree has a branching factor of 5. KIPDA can send 9 decoy values before it uses more energy than hop-by-hop aggregation that uses IDEA or RC5 encryption sending one value.

where b the number of bits in a block or segment (this equation only assumes one block is used), Dec_T and Enc_T are the times to encrypt and decrypt b bits, and BW is the bandwidth, which for both architectures is 0.25 bits per microsecond. Since bandwidth is the same for both architectures, the results are the same. The equation to determine the time for KIPDA is given as:

$$T_{KIPDA} = (c+1) \cdot \frac{m \cdot b'}{BW}. \tag{4.25}$$

The results are shown in Figure 4.9. KIPDA is compared to the time it takes for hop-by-hop aggregation using IDEA, RC5, and RC4 encryption. 10 bits per value are used and the network branching factor is 5. The analysis shows that KIPDA is time efficient, and can process 47 decoy values before it reaches the same time used

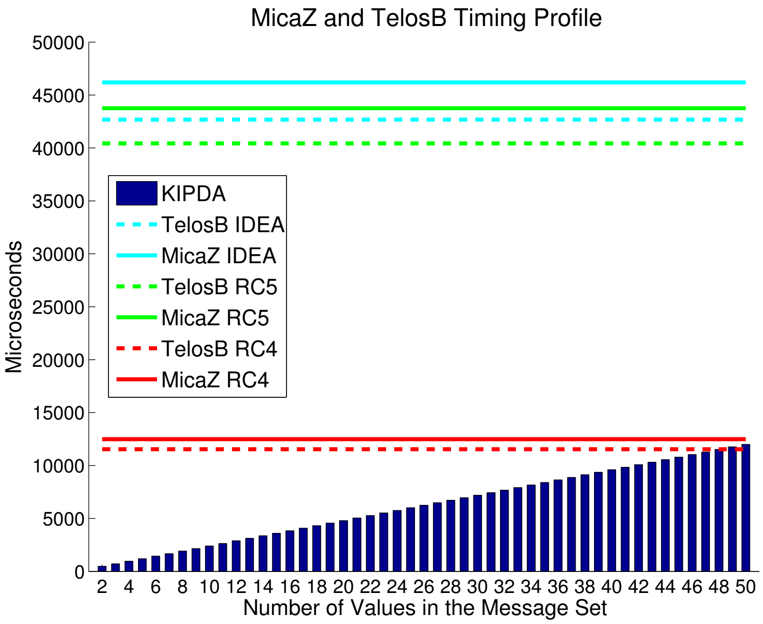


Figure 4.9: The time delay of KIPDA versus hop-by-hop aggregation that uses the IDEA, RC5, and RC4 ciphers. MicaZ and TelosB architectures are shown. KIPDA can send 46 decoy messages before any other encryption technique on either architecture can perform the aggregation process.

by a node performing hop-by-hop aggregation with RC4 processing one value. For IDEA and RC5 encryption, KIPDA can process about 160 decoy messages before it uses the same amount of time these schemes use. If delay is intolerant, privacy is a concern, and energy consumption needs to be conserved, KIPDA could consume more energy by transmitting more camouflage values to achieve these three goals.

4.6 Summary of Chapter

Because KIPDA hides sensitive data in plain sight, data aggregation is easily and efficiently computed, and the in-network processing delay can be reduced compared to hop-by-hop encryption methods. I have shown that KIPDA uses less energy than hop-by-hop encryption even though more camouflage messages are communicated. I quantified the energy efficiency of the proposed method in terms of the amount of camouflage data used and studied the trade-offs between the protocol's effectiveness and its resilience against collusion and capture attacks. It is possible to conserve energy and protect privacy by transmitting more information over a node's radio, if decoy values are used strategically. The next chapter confirms the energy analysis of this chapter through simulations and implementations.

Chapter 5

Simulations and Implementations

This chapter presents simulations and implementations of the algorithms from the previous two chapters. This includes simulations of MDNSs in MATLAB and KIPDA in PowerTOSSIM-Z, in addition to implementations of MDNSs on Android OS smart phones and KIPDA on Moteiv T-mote Invent sensors. Possible applications include detecting radiation in a city, perturbing continuous data sets for privacy-preserving data mining, or protecting medical data while monitoring simultaneously several patients. The simulations and implementations illustrate the effectiveness and feasibility of the two protocols, how they can be adapted to real world applications, and the trade-offs between energy and privacy. I study how MDNSs can perform given a limited number of participants, how accurate they are given this limitation, and how much energy is saved when encryption is eliminated. I also study how advantageous it is for KIPDA to spend extra energy transmitting decoy messages.

5.1 MDNS MATLAB Simulations

MATrix LABoratory or, MATLAB, is an environment for numerical computing [111]. As of 2004, it had over one million users in research and industry [74]. Because it is popular, easy to use, and a supports of a wide range of functionality, it was chosen to simulate MDNSs. The first MATLAB simulation demonstrates how to detect possible radiation threats in a city. The second simulation illustrates how MDNSs can be applied to continuous data, which is compared with a popular privacy-preserving data mining technique, random data perturbation. Each simulation is enhanced with DA, showing the trade-offs between utility gained and privacy lost.

5.1.1 Cell Phone Radiation Threat Detection

Participatory sensing could potentially help detect and locate radiation threats in a city, such as the detonation of a dirty bomb, loss of radioactive medical material, or spread of radiation from a nuclear reactor accident. In this scenario, I assume that cell phones are equipped with radiation monitors and GPS devices. Locations are quantized into different quadrants, each with a unique label. I also assume that individuals care about the privacy of their locations. With reasonable parameter assumptions (number of locations, number of discernible radiation levels, and number of participants), MDNSs can maintain location confidentiality and identify locations containing radiation threats, if they exist.

Cell phones are ideal for radiation detection, and the United States Department of Homeland Security has considered their use [68]. If radiation sensors were installed at fixed locations, they might be tampered with or avoided, which is more difficult with cell phones because they are owned by many independent individuals. As an incentive to promote participation, aggregate information could be disseminated freely to participants. Since readings from an individual cell phone might not be as accurate as the combined readings from a larger population, access to aggregate information would be advantageous. For an event such as the Fukushima Daiichi nuclear accident, participants might prefer to send the unperturbed data and receive more accurate readings. Either way, in such a situation, immediate feedback would be beneficial, especially to determine if radiation has spread further than publicly acknowledged.

Simulation Setup

Before I explain the simulation setup, I give a small example of a geographic area divided into a 3 × 3 grid, shown in Figure 5.1. The total population of cell phones (participants) is 450,000 and is equally divided among the 9 locations. In the actual simulation, I do not assume a uniform population distribution and instead follow a more realistic model given by Bertaud et al. [17]. I simulate three radiation levels: low, medium, and high. Depending on the level of radiation, each location's distribution of reported levels will be shifted lower or higher. For example, in Figure 5.1, location 6 contains a *threat distribution*, illustrated by the black histogram. This distribution, exponentially shifted towards the higher range, contains 28,571 high radiation readings, half that number (14,286 medium radiation readings) in the medium radiation level, and 7,143 readings that are low. Benign locations, characterized by the *non-threat distributions*, are shown in black in locations 1-5 and 7-9. These distributions are skewed in the reverse direction: 28,571 participants with low readings, etc. Figure 5.1 also shows the reconstructed distributions in light grey, which resemble the original distribution closely enough that important decisions could be made such as where to send response teams.

San Francisco, which has roughly 46.7 square land miles, is used as an example city. I chose the number of distinct locations to be 48, which works well with DA due to its high number of composites. A hexagonal grid was used where each location

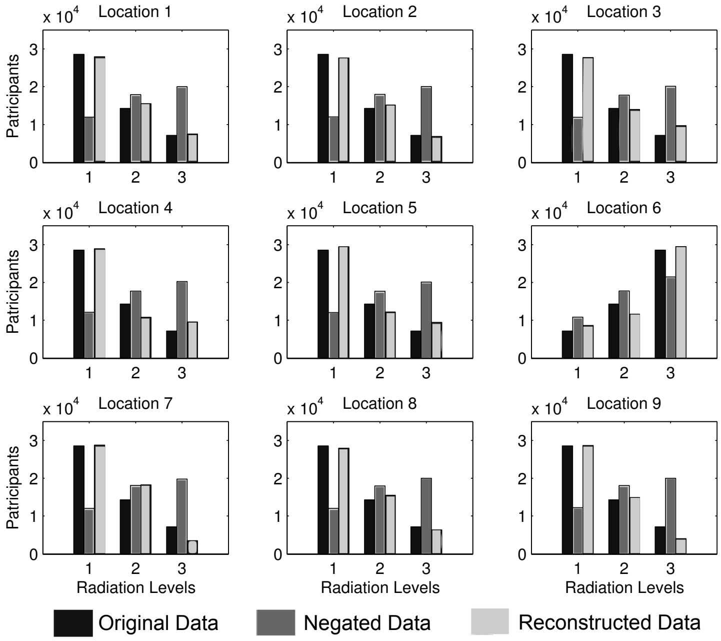


Figure 5.1: Histograms for a multi-dimensional negative survey of 9 locations and 3 radiation levels. The y-axis measures the number of participants per level of radiation (the x-axis). Location 6 is suspicious since its radiation levels form a threat distribution. The other locations have non-threat distributions.

covers roughly one square land mile. This size would allow a response team with more powerful equipment (such as helicopters equipped with radiation detectors) to pinpoint the exact location of a threat.

San Francisco has a population of about 815,000, therefore, I varied the number of participants from 100,000 to 500,000 in increments of 100,000 to study the effect of the number of participants on utility. 500,000 is a good conservative maximum estimate of the general population willing to participate [39]. As mentioned earlier, the spatial distribution of participants is assumed to follow a standard urban model taken from Bertaud et al. [17] where the population is concentrated at the central business district and is gradually reduced further from this center.

Radiation levels were divided into 3 categories, course graining continuous values into one of the categories. While experiments show that more categories would increase the granularity of the data, accuracy improves with fewer radiation levels (Chapter 3, Section 5.3). However, if there were only 2 radiation levels, privacy would be lost and adversaries could determine a participant's location, if a threat existed at a single location and the participant is at that location.

Each participant's cell phone samples the environment for the radiation level and notes its location. It then perturbs this information according to Chapter 3, Section 2.1 and sends the perturbed values to the base station. After the base station collects the perturbed data (one sample from each cell phone), it reconstructs the original distribution according to the protocols from Chapter 3, Section 2.2.

The base station determines if a threat exists by computing the linear regression at each location from its reconstructed histogram of radiation levels, assuming that histogram bins are one unit apart. Figure 5.2 illustrates this technique. Ideally, a location reporting elevated radiation levels will have a positive slope from the linear regression, and a location with a typical distribution will have a negative slope. Yet, this is not always the case. I chose the actual slope thresholds to minimize the overall number of type I and type II errors. Thresholds could be adjusted to favor one error type over another. For example, one strategy might send response teams to investigate false positives, rather than allowing a false negative to slip through. I chose the thresholds *a posteriori*, but in a real deployment these values would be chosen *a priori*, with additional domain knowledge.

I ran the simulation 1000 times for each increment of participants, assigning the threat distribution to a random location in 500 of the runs. In the other 500 runs I assigned a non-threat distribution to all locations.

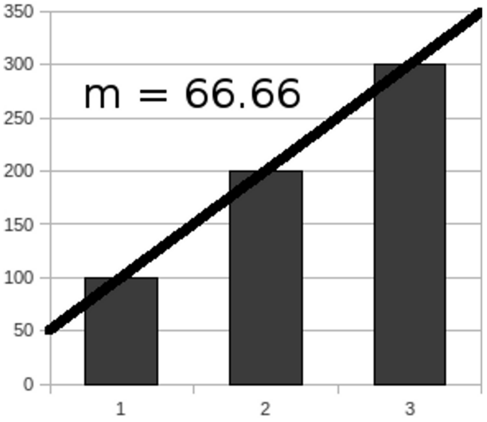


Figure 5.2: Example of the technique used to determine radiation threats with the slope of linear regression with a histogram.

Results and Analysis

Table 5.1 summarizes the results, showing the number of false positives and false negatives, and accuracy. Accuracy is the percentage of true positives that correctly determined the threat location. The average privacy and utility values (defined in Chapter 3, Section 3) are also given. Figure 5.3 displays the same results graphically.

Because accuracy was low for a single dimension, shown in the first four rows of Table 5.1, I next applied DA from Chapter 3, Section 5.1. The other rows in Table 5.1 show results for various DA settings. The location dimension of 48 categories was factored into 2 dimensions of 6 and 8 categories; 3 dimensions of 4, 4, and 3 categories; and 4 dimensions of 2, 2, 4, and 3 categories. With 4 dimensions, I obtain 100% accuracy with 200,000 or more participants.

In summary, radiation monitoring with cell phones would be practical for an example city such as San Francisco, with 46 square land miles and a population around 900,000, even if only 200,000 people participated. Using the reconstruction Algorithm 3.3, I conclude that MDNSs with DA can accurately determine if a radiation

Table 5.1: Results of the cell phone radiation detection simulation. Each test consisted of 1,000 runs, 500 runs contained a radiation threat in a random location, and another 500 runs contained no threat.

Samples	False Neg.	False Pos.	Acc. of True Pos. % (Ratio)	Avg. Privacy	Avg. Utility
1 locational dimension with 48 categories					
100,000	246	246	5.5 (14/254)	0.0282	4.54E-04
200,000	244	245	7.8 (20/256)	0.0252	2.27E-04
300,000	244	244	18.0 (26/256)	0.0241	1.51E-04
400,000	241	242	18.9 (49/259)	0.0234	1.13E-04
500,000	238	239	19.9 (52/262)	0.0230	9.08E-05
2 locational dimensions with 8x6 categories					
100,000	246	248	11.8 (30/254)	0.0350	1.90E-04
200,000	250	251	21.2 (53/250)	0.0319	9.48E-05
300,000	222	222	31.3 (87/278)	0.0307	6.32E-05
400,000	199	199	39.2 (119/301)	0.0300	4.74E-05
500,000	194	194	44.4 (136/306)	0.0296	3.79E-05
3 locational dimensions with 4x4x3 categories					
100,000	203	205	56.6 (168/297)	0.0586	3.09E-05
200,000	139	139	81.7 (295/361)	0.0554	1.54E-05
300,000	87	87	92.5 (382/413)	0.0542	1.03E-05
400,000	58	58	94.3 (417/442)	0.0535	7.71E-06
500,000	37	37	98.3 (455/463)	0.0531	6.17E-06
4 locational dimensions with 2x2x4x3 categories					
100,000	17	17	99.4 (480/483)	0.1444	4.41E-06
200,000	0	0	100 (500/500)	0.1411	2.20E-06
300,000	0	0	100 (500/500)	0.1398	1.47E-06
400,000	0	0	100 (500/500)	0.1392	1.10E-06
500,000	0	0	100 (500/500)	0.1387	8.81E-07

threat exists and where.

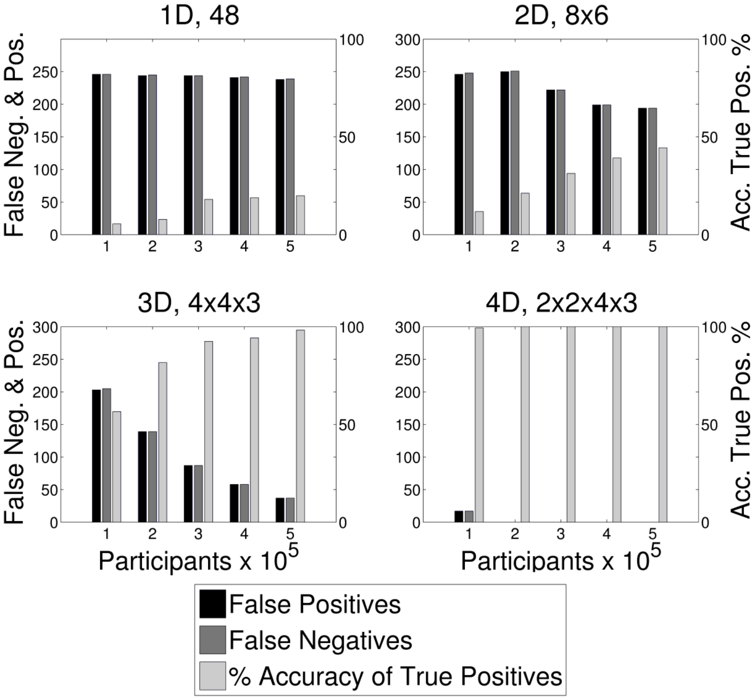


Figure 5.3: Results from Table 5.1 displayed graphically. (Top left) Location is treated as a single dimension of 48 categories. (Rest) Locations are dimensionally adjusted to 2, 3 and 4 dimensions, with the number of categories given for each dimension.

5.1.2 Reconstructing Continuous Values [1]

In addition to categorical data such as locations and radiation levels, MDNSs could be applied to continuous data such as temperature or humidity. I reconstruct the probability density functions of different underlying distributions and compare the parameters of these distributions to the original parameters. This application is potentially relevant in privacy-preserving data mining as an alternative to random data perturbation [9].

[1] The idea for this application was suggested by Benjamin Edwards.

Algorithm 5.1 MDNSs on Continuous Data

```
                                          ▷ "-" denotes set difference.
                                          ▷ || denotes concatenation.
                ▷ Leading zeros can be applied to sensed value $SV$ if needed.
for each numeral $n_i$ in sensed value $SV$, where $SV = n_1||n_2||\ldots||n_x$ do
    $m_i \leftarrow$ Select from $\{0, 1, \ldots, 9\} - \{n_i\}$ with uniform probability.
end for
return $m_1||m_2||\ldots||m_x$
```

Simulation Setup

Any fixed point number can be represented as a collection of categories by labeling each digit's position (1's, 10's, 100's,...) with a value ranging from zero to nine. Thus, a fixed point number with n digits can be treated as an n dimensional negative survey, with each dimension having ten potential categories; it is then straightforward to apply the protocols presented previously, as illustrated in Algorithm 5.1.

Samples were generated from the following two probability distributions, normal and exponential, and rounded so that they contained only 2 and 3 significant digits:

$$\mathcal{N}(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{\sigma}}, \qquad \mathcal{E}(\mu) = \frac{1}{\mu} e^{-\frac{x}{\mu}}. \tag{5.1}$$

Any distribution could be used. These two were chosen because of their ease to determine their parameters from a collection of samples. Tests used $\mathcal{N}$(500, 100) and $\mathcal{E}$(100), and the tails of the distribution were truncated at 0 and 1000. The number of samples for the two distributions was varied from 1,000 to 9 billion in exponential increments.

The base station reconstructed the frequency of each number of the D most significant digits (the probability density function) of the underlying data according

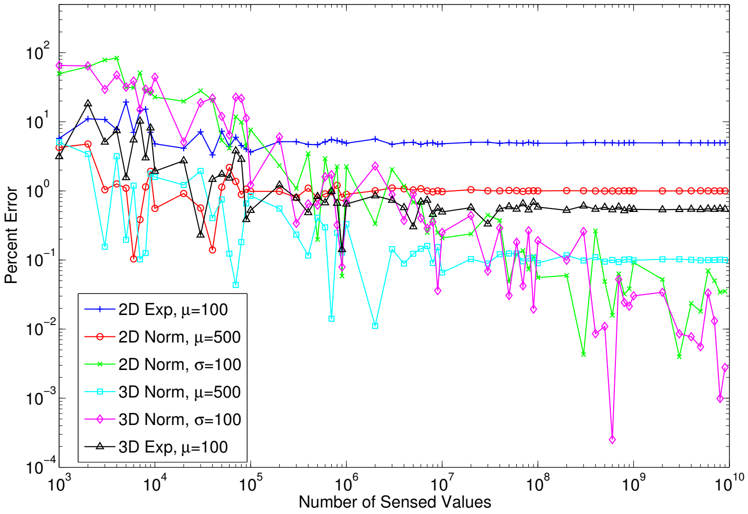


Figure 5.4: Parameter reconstruction error from the continuous negative survey simulation measured as a percentage difference from the original parameter. Each data point is an average of 20 runs.

to the protocols in Chapter 3, Section 2.2. Distribution parameters were determined for the reconstructed data set, A, using a maximum likelihood estimate. They were then compared to the original parameters used to construct the data set X.

Results and Analysis

The percentage error was calculated from the difference between the estimated and original parameters, divided by the original parameter. Each data point is an average of 20 runs. Figure 5.4 shows these results. It suggests that the maximum achievable accuracy depends on the parameter and its value, the distribution, and the number of dimensions. All parameters are within 5% of the original parameter values after 200,000 sensed values.

Comparison of MDNSs on Continuous Data to Random Data Perturbation

Random data perturbation (RDP) algorithms proposed by Agrawal, et al. [9] and later Zhang et al. [155] perturb continuous data by adding noise drawn from a known distribution. RDP adds a randomized value, r_i, drawn from the known distribution over a finite range, to a datum, x_i. The perturbed data is then reconstructed to an approximation of the original data using an iterative algorithm based on Bayes Theorem. Reconstruction uses an expectation maximization (EM) algorithm [43] that provably converges to the maximum likelihood estimate of the original distribution [7]. RDP techniques assume that the data and the noise are drawn from a continuous domain.

My preliminary research [85] in collaboration with James Horey compared the original negative survey method to a modified categorical RDP technique. However, the accuracy of the comparison was compromised by the modification. Here, I compare MDNSs on continuous data to the original RDP technique.

The original RDP algorithm proposed by Agrawal and Srikant [9] was implemented with a triangle, plateau, and step distribution. These distributions test the ability of RDP and MDNSs to reconstruct discontinuous functions. A moving average of size 7 was used on the reconstructed distributions from the negative surveys to help smooth the data. However, with enough participants this is not necessary.

MDNSs and RDP each have strengths and weaknesses. MDNSs require a large number of samples to reconstruct the probability density function (PDF) accurately. For example, in Figure 5.5, with 10^8 samples the root mean square error (RMSE) of RDP for the triangle distribution is 419,732.8, while with MDNS it is 227,941.2, a 45.7% improvement. For the plateau distribution with the same number of participants, the RDP has a RMSE of 63,521.3, while MDNS has 24,153.6, a 62.0%

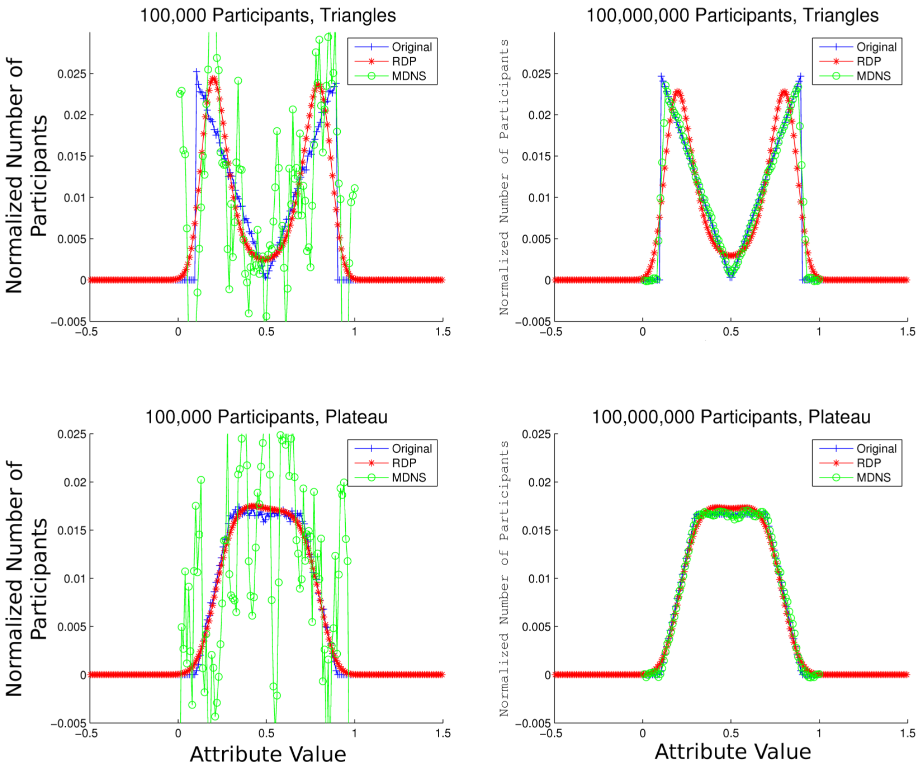


Figure 5.5: Comparison of RDP to MDNS. The PDFs of RDP (red) and MDNS (green) are compared to the original PDF (blue).

improvement. However, with 10^5 samples, MDNSs could not reconstruct the PDFs well enough to distinguish between the two distributions. MDNSs handle discontinuous PDFs and PDFs whose derivatives are discontinuous better than RDP as illustrate in Figure 5.6. In addition, the stopping criteria for RDP is problematic [9], and the running time is considerably larger than MDNSs.

In summary, I have shown that when approximately 22% of San Francisco's population participates in a MDNS radiation detection scenario, radiation threats can be determined accurately inside the city to within a square mile. This is accomplished while maintaining individual location privacy and the conservation of energy in the

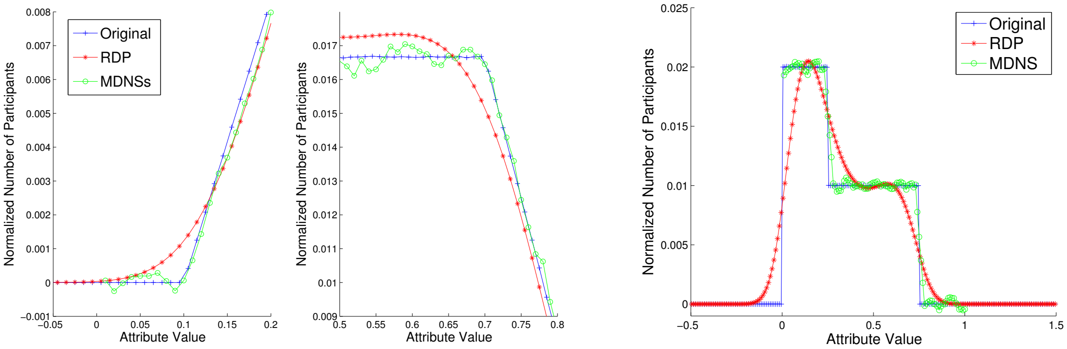


Figure 5.6: Comparison of MDNSs to RDP. MDNSs outperform RDP on discontinuous PDFs (two left panels), and PDFs whose derivatives are discontinuous (right panel).

participant's cell phone because of the elimination of encryption, key management, and key distribution. MDNSs on continuous data provide better reconstruction of perturbed data sets if the number of samples are large enough. They can also reconstruct PDFs with discontinuities and PDFs with derivatives that have discontinuities more accurately than RDP.

5.2 KIPDA PowerTOSSIM-Z Simulations

In this section, KIPDA is simulated on TOSSIM [103] with the aid of AVRORA [142] and PowerTOSSIM- Z [122]. Attention is given to the amount of energy consumed, comparing KIPDA to hop-by-hop encryption aggregation using various encryption ciphers.

TOSSIM, or TinyOS Simulator, is a discrete event simulator of sensor networks running the TinyOS operating system which can scale easily to thousands of simulated mote nodes. Developers can test, debug, and analyze their algorithms directly from code written for TinyOS in a controlled and repeatable manner. TOSSIM (version 2.1.1) was chosen for this simulation because it is readily available, widely used, and can easily be configured for different topologies and noise models.

PowerTOSSIM-Z measures the amount of energy consumed by each node in the TOSSIM simulator. It outputs debugging information which is parsed to obtain the energy consumption per component per node in the network. These debugging statements are generated each time a component changes power states.

KIPDA and hop-by-hop encryption aggregation were simulated in TOSSIM on the MicaZ architecture, the only architecture in TinyOS 2.X that TOSSIM supports. MicaZ is the architecture of a prototypical sensor mote, used by hundreds of research groups. It contains an ATM128 micro-controller that runs at 7.37 MHz, at least 128KB of AT45DB flash memory, and a CC2240 radio chip that can run at a max throughput of 250 Kbps, but realistic throughput can be half this amount [47].

5.2.1 Simulation Setup: Design Decisions

In this section, I describe seven steps the design decisions used to create the TOSSIM simulations.

1. The first step implemented a base station querying (or disseminating) data to each node. I used the TinyOS Enhancement Proposal (TEP) 118 for the dissemination scheme because (1) it is already implemented and (2) many of the design decisions are decided by these protocols. Since each test uses the same underlying protocols (TEPs) to disseminate information, the conditions for each simulation are the same.

2. In the second step, each node sends a response to the base station's query without aggregation. I used the TEP 119 collection scheme, which is based on the Collection Tree Protocol (CTP) (TEP 123). CTP does not promise 100%

reliability. “It is a best effort, but a best effort that tries very hard.” [118]. Duplicate packets were detected and dropped in the simulations. Creating a technique to detect and retransmit dropped packets would involve higher energy costs due to the extra network protocols, and would make direct algorithmic comparisons difficult.

3. The third step aggregates the responses from the second step according to a scheme similar to Tiny AGgregation (TAG) [110]. Timing is critical in aggregation. A node cannot send its aggregate too early or it will miss some incoming information. If it waits too long, its parent may have already sent its aggregate. Hence, each node must wait a specific amount of time. I base this time on the number of hops a node is from the root node. This information is recorded with 16 bits in the dissemination's protocol beacon. Each node then waits according to the following equation:

$$x \cdot (\text{Max Hop Level - Current Level}) \tag{5.2}$$

where the unit is in seconds and x varies between 1 and 2. These values for x give each node plenty of time to collect its children's information.

The sum aggregate was used to test this step. Each node reported the value 1, hence, the network aggregate would reflect the number of nodes in the network. The correct aggregate was reported 98% of the time. Since radio noise levels are consistent between all runs, and the same number of bytes are communicated between KIPDA and hop-by-hop encryption, dropped packets should not affect the simulations' output.

Programming the aggregation was simplified by using CTP, which contains a special function to intercept data en route to the base station. This intercept function determines whether a message is forwarded or not (by returning true or false respectively). Aggregation packets always return false (as opposed to packets which report voltage use for each node). The information from an aggregate message is stored in the node where it is aggregated with other received information until a timer determines to send the combined aggregate to the next hop.

4. The fourth step implemented PowerTOSSIM-Z to determine the amount of energy used in the third step. This was relativity straightforward with one exception: how the radio draws power. The scripts for PowerTOSSIM-Z assume the radio is always in receive mode, unless it is transmitting. One way to stop the radio is through a system call to the operating system. However, when the radio is off, it cannot receive any information, including beacon information from the lower network protocols. Low power listening (LPL) [94] suggests a solution by having the radio sleep in a very low power state, waking after a random amount of time to listen for a packet. If the radio does not hear anything, it goes back to sleep. However, LPL is not implemented in TOSSIM or PowerTOSSIM-Z. Therefore, I made two assumptions about radio usage:

    - Since TOSSIM cannot distinguish between the lower transmit power levels, I assumed all radio transmissions occur at the highest power level.
    - I assume a perfect LPL mode where power is drawn only when the radio is broadcasting or receiving. This is an unrealistic ideal mode because it assumes that the radio magically wakes up exactly when it needs to receive a message.

5. The fifth step implemented hop-by-hop encryption in the network. The following encryption ciphers were used: AES [114], RC5 [132], Trivium [41], SkipJack [117], and TinyECC [106]. AES is a block cipher technique that operates on blocks of 128 bits. RC5 uses variable key and block sizes, but for this research, the blocks are 64 bits. SkipJack is a also block cipher with 64 bit blocks. Trivium is a stream cipher that works on segments of 16 bits, and TinyECC

Method	Optimized Assembly?	Type	Size of block or stream (bits)	Size (bytes) ROM/RAM
AES	No	blocking	128	5,156/2,384
RC5	No	blocking	64	2424/60
RC5	Yes	blocking	64	3434/60
Trivium	No	steaming	32*	8366/108
SkipJack	No	blocking	64	4458/316
TinyECC	No	asymmetric	648	24,232/2,019
TinyECC	Yes	asymmetric	648	24,952/1,990

Table 5.2: Various forms of encryption ciphers and their properties. *Trivium is a streaming encryption technique and operates on segments of 16 bits, but needed an extra 16 bits to determine its order in the stream.

is an asymmetric encryption scheme designed for resource-constrained devices which uses 648 bits for its ciphertext. Table 5.2 gives the encryption technique and their stream or block sizes. Optimized assembly code instructions existed for RC5 and TinyECC on the MicaZ architecture and were included in the analysis.

There were two complications involved with this step:

- Trivium was treated as a special case because it is a streaming encryption technique. Stream segments are 16 bits. However, I discovered from the simulations that Trivium needs to transmit an extra 16 bits to denote the ciphertexts' positions in the encryption stream. Ciphertexts must be encrypted and decrypted in the same order because these primitives keep a running state. WSNs are known to drop or send duplicate packets. Hence, a mote that receives a ciphertext without its predecessors can decrypt null information as many times as needed before the ciphertext is decrypted. Precautions were added to the simulations to handle out of order Trivium segments.
- Second, PowerTOSSIM-Z did not initially measure the time spent en-

	Encrypt Primitive	Decrypt Primitive
Method	CPU Cycles / ms / μJ	CPU Cycles / ms / μJ
AES	7,816 / 1.061 / 24.07	11,288 / 1.532 / 34.76
RC5	1,534 / 0.2081 / 4.72	2,215 / 0.2883 / 6.543
Trivium	3,353 / 0.4550 / 10.32	2,672 / 0.3626 / 8.22
SkipJack	2,533 / 0.3437 / 7.78	2,935 / 0.3982 / 9.03
TinyECC	25,600,393 / 3,470.0 / 78,821.0	19,538,454 / 2,650.0 / 60,157.0

Table 5.3: Encryption/Decryption cost in CPU cycles, microseconds, and microjoules reported from the AVRORA simulator for 5 encryption ciphers. Costs are given for encrypting and decryption one block or stream segment whose size is given in Table 5.2

crypting on a mote's microcontroller. This complicated careful measurement of energy usage. I addressed this problem by using an AVR microcontroller simulator, AVRORA [142], which fully emulates the microcontroller produced by Atmel that runs on the Mica2 and MicaZ architectures. AVRORA recorded the time it took to run each encrypt/decrypt primitive 1,000 times for the AES, SkipJack, RC5, Trivium, and TinyECC ciphers. The average was then taken for each primitive. The results are shown in Table 5.3 and were incorporated into PowerTOSSIM-Z.

6. The sixth step implemented KIPDA. This was straightforward, given the experience from the previous five steps. Emphasis was given to energy use and the size of the message vector, rather than implementing all possible security enhancements.

7. The final step involved comparing hop-by-hop encryption to KIPDA. I used the three different network topologies given in Figure 5.7. For each topology, each encryption cipher was compared to KIPDA that used the corresponding number of bytes sent over the radio, e.g., AES's block size is 128 bits, hence it was compared to KIPDA that sent 16 bytes of data over the radio. Each

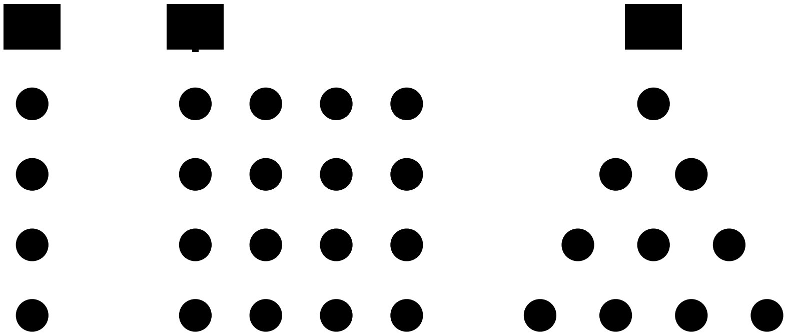


Figure 5.7: Example of three different topologies: chain, grid, and tree, used to test KIPDA against hop-by-hop encryption. Squares represent the base stations and circles represent the nodes. 49 nodes were used in all simulations.

test ran 40 times for 10,000 virtual seconds. When x=2 in Equation (5.2), the maximum number of hops from the base station is set at 13, and the base station waits an additional 4 seconds, an aggregation epoch took place every 30 virtual seconds. This gave a total of approximately 333 aggregations per simulation. Because this is a small number, small differences in Figures 5.8, 5.9, and 5.10 will be magnified over the complete lifetime of an actual network.

5.2.2 Simulation Results

Figures 5.8 reports the total energy used in the network for chain, grid and tree topologies for each encryption cipher in hop-by-hop aggregation, and is compared to KIPDA configured to send the same amount of information over the radio. TinyECC is not represented because it uses a significant amount of energy. From Figure 5.8 it is evident that KIPDA conserves energy. In addition, it protects sensitive information from in-network nodes, which hop-by-hop encryption does not. Figure 5.9 gives the average energy used per node. While network-wide energy use has little variance, each node's energy use varies significantly depending on its location. There has been much research into the maximum lifetime routing problem in WSN [30], yet, this is beyond the scope of this dissertation.

The results show that KIPDA conserves energy when broadcasting the same amount of information over the radio compared to hop-by-hop encryption. I next determine how many messages KIPDA can send before exceeding the energy used by hop-by-hop aggregation for various encryption ciphers. A *threshold ratio* was defined to compare energy use between the two schemes. The denominator represents hop-by-hop encryption, while the numerator represents KIPDA. Values higher than one indicate that KIPDA uses more energy, while values lower than one indicate that KIPDA saves energy. The threshold ratio is given as follows:

$$\frac{Energy[Tx(l \text{ bits}) + Rx(l \text{ bits})]}{Energy[Tx(m \text{ bits}) + Rx(m \text{ bits}) + Enc(m \text{ bits}) + Dec(m \text{ bits})]} \tag{5.3}$$

where Tx is transmit, Rx is receive, Enc is encrypt, Dec is decrypt, and $Energy$ is the energy used to perform these tasks. This ratio predicts and quantifies how many decoy messages it takes to either conserve or waste energy in the network compared to hop-by-hop encryption. Figure 5.10 illustrates when Equation (5.3) is set to one, the same amount of energy is used between KIPDA and hop-by-hop encryption aggregation.

The following justifies the threshold ratio. Given a network with N nodes, there will be N aggregation messages sent by each node. Each message sent must also be received by another node. The size of the message vector for KIPDA is l bits, while for hop-by-hop encryption it is m bits. In hop-by-hop encryption, each packet must also be encrypted and decrypted. This ratio measures the total network use, and as a caveat, cannot predict any single node's energy use.

The following can be observed from the simulation results:

- The threshold ratio predicts for most forms of encryption the number of bytes

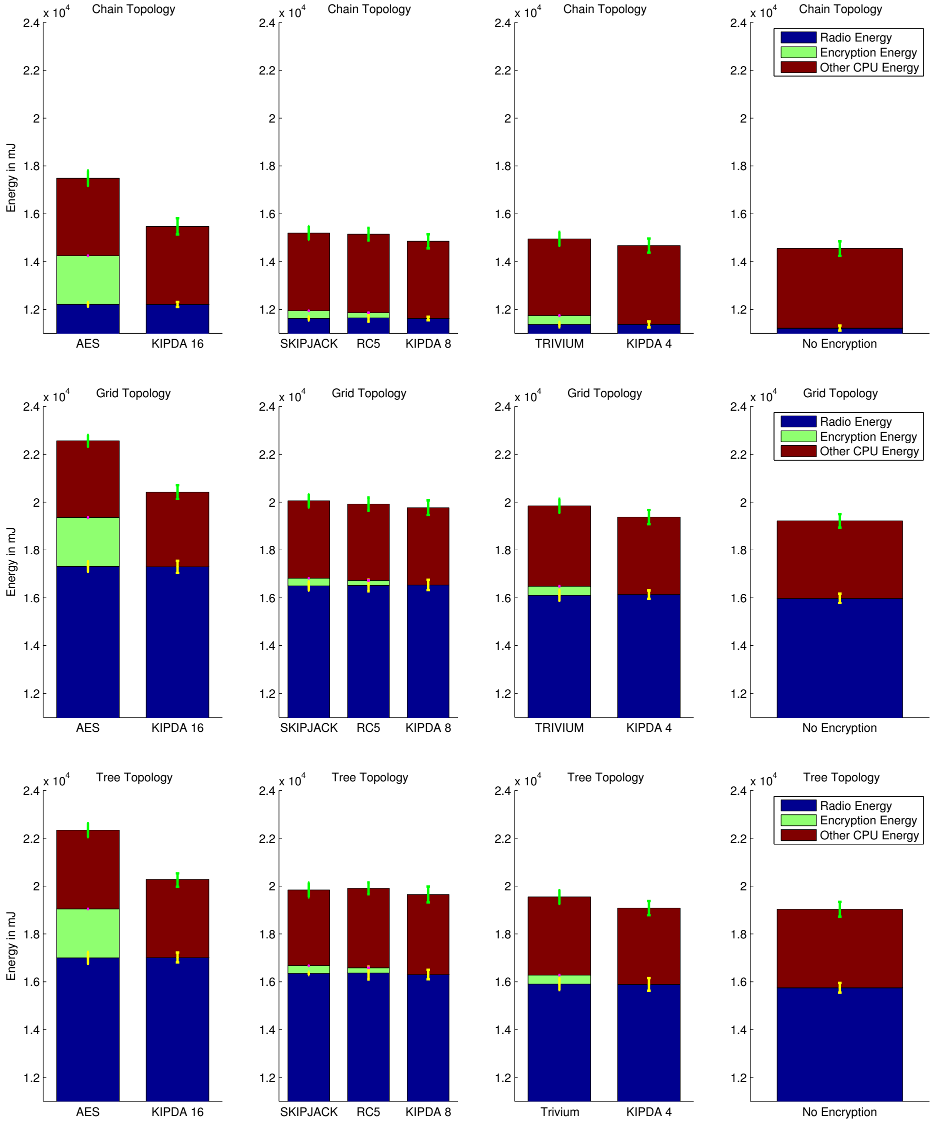


Figure 5.8: Total energy consumption for a WSN of 49 nodes using KIPDA MAX/MIN aggregation and hop-by-hop aggregation using AES, SkipJack, RC5 and Trivium encryption for three topologies from Figure 5.7. 333 Aggregations were performed. The number after KIPDA indicates the number of bytes for a message vector.

that KIPDA can use and still consume an equivalent amount of energy to that of hop-by-hop encryption. However, KIPDA with 33 messages appears to use

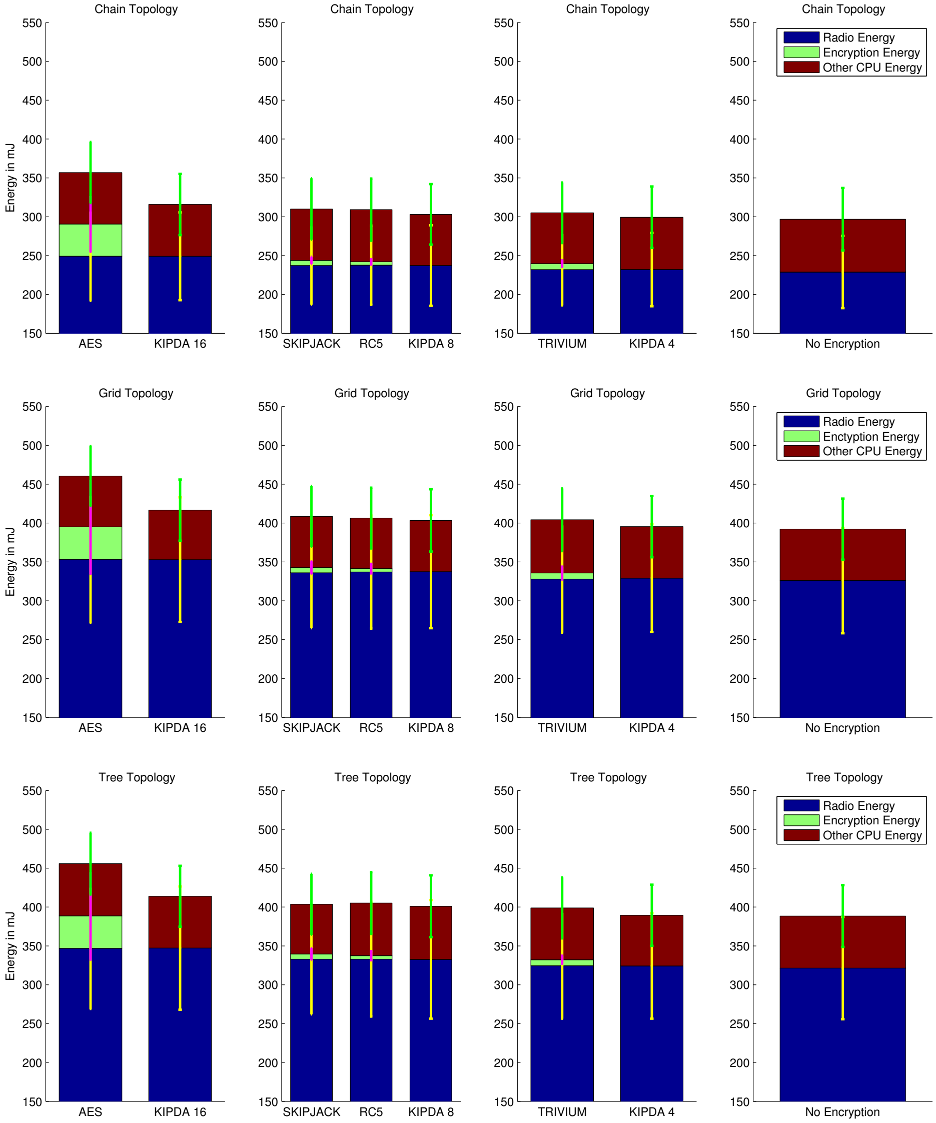


Figure 5.9: Average energy used per node in a WSN using KIPDA MAX/MIN aggregation and hop-by-hop aggregation using AES, SkipJack, RC5, and Trivium encryption for three topologies from Figure 5.7. 333 Aggregations were performed. The number after KIPDA indicates the number of bytes for a message vector.

more energy than hop-by-hop AES encryption. This could be attributed to packet loss from transmitting 33 bytes of data. Because of the larger packet

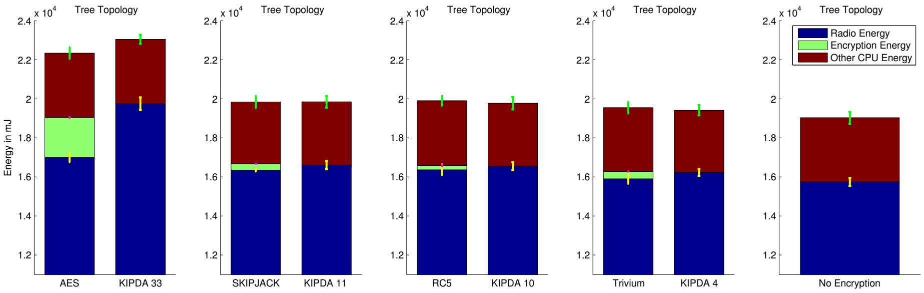


Figure 5.10: Comparisons of total network energy used between hop-by-hop encryption aggregation and KIPDA MAX/MIN aggregation where the threshold ratio equals one. The number after KIPDA indicates the number of bytes for a message vector.

size, there is a higher probability that at least one packet would be dropped due to noise in the environment. Retransmitting these dropped packets could account for the extra energy use.

- The standard deviation is low for the total encryption energy used in the network. The nodes themselves vary in their energy use, which is due to their individual location in the network, e.g. closer or further to the base station.

- The results show that, as expected, a considerable amount of energy is spent on the radio. This is due to the underlying dissemination and collection protocols which often transmit beacon packets to discover the best routes.

- Sometimes it takes half as much energy to encrypt and decrypt a message as it does to send and receive the message. For example, on the MicaZ architecture it takes 24.07 microjoules to encrypt and 34.76 microjoules to decrypt using AES, and 53.45 and 60.31 microjoules to send and receive the messages respectively, assuming the average bandwidth of the MicaZ architecture is one half of the maximum bandwidth.

- It is clear where KIPDA gets its energy advantage. AES uses 17,000 and 2,046 microjoules in the radio and encryption respectively. KIPDA takes advantage of the eliminated 2,046 microjoules to communicate up to 33 bytes of data.

- Because of the timing inefficiency of the underlying dissemination and collection protocols, KIPDA could not capture timing information. Future work can focus on delay intolerant networks where timing is critical, such as the work by Quan et al. [129].

I have shown that the elimination of encryption in KIPDA allows extra energy to be spent in the radio, allowing a modest amount of camouflage data to be transmitted. This modest amount of data conserves overall energy use with k-indistinguishable privacy.

5.3 Implementations on Physical Devices

The goal of this section is to measure the energy savings of the MDNSs and KIPDA protocols on actual physical devices. MDNSs were implemented on Android smart phones and two experiments were performed. The first experiment determined how many more messages can be transmitted compared to messages that are encrypted with the Secure Sockets Layer protocol. The second experiment illustrated the functionality and feasibility of MDNSs. It tested whether smart phones following the MDNS protocol that sample their microphone can determine noisy locations on a university campus.

KIPDA was implemented on Moteiv T-mote Invent sensors. Three experiments were performed to determine that KIPDA, transmitting 18 bytes per message vector, uses less energy than hop-by-hop aggregation with the AES cipher transmitting 16 bytes per message. First, I measured the time it took to drain the battery for each method. Second, I measured the voltage level of each sensor after 10 aggregations.

Third, I measured the time for the AES encrypt and decrypt primitives. Collecting energy information from the Moteiv T-mote Invent sensors is challenging. While voltage information could be obtained, current level could not be determined due to the limitations of the devices. The state of the art solution would require electrical engineering expertise. While none of these three methods are perfect, they do not disprove that KIPDA can conserve energy when it transmits extra information over the radio.

5.3.1 MDNSs on Android Smart Phones

Implementation Setup

The MDNS node protocol was implemented on Android OS smart phones. Two experiments were performed:

- The first experiment tested the energy cost of communicating data with and without the Secure Sockets Layer (SSL) encryption. SSL was chosen because it is popular and easily implemented on smart phones. In the first test, messages were sent using SSL until the battery was drained to half its fully charged level. The phone was then recharged and the experiment repeated without SSL. To maintain uniform experimental conditions, the phone stayed at the same location, all other non-essential applications were regularly closed every 15 minutes, and experiments were performed in the middle of the night to prevent unwanted calls or text messages.
- The second experiment posed the task of identifying the noisiest locations on the University of New Mexico's campus. Three smart phones were programmed to collect sound samples, approximately 7,300 in total. The three participants carried their phones throughout the campus, spending about the same amount

> of time in each different location. Latitude and longitude were divided into 4 and 6 categories respectively for a total of 24 different locations. Because of the limited number of samples, longitude was dimensionally adjusted to two dimensions of 2 and 3 categories. Sound was sampled from the phone's microphone and quantized into three different volume levels. Volume levels were calibrated to 3 different categories corresponding to a room with normal conversations, outside in a quiet environment, and outside next to rush hour traffic. Since the campus is surrounded on 4 sides by busy streets, I expected the highest noise from the perimeter locations. The phones sampled volume and location, perturbed the information, and sent it back to a base station implemented in Java.

For both experiments, location was obtained from GPS satellites. Although this consumes more energy than other methods such as multilateration of radio signals between cellular towers, I required precise location information because the experiments were performed in a small geographic area.

Implementation Results

For the first experiment, the base station was able to receive 8,612, or 16.22% more messages without encryption, than the 7,401 messages that used SSL. It took 159.2 minutes to send the messages with SSL, and 143.6 minutes to send the messages with no encryption. This implies that approximately 4.836 more milliseconds was spent per message using SSL, which involves exchanging symmetric keys with asymmetric encryption, encrypting the data with the RC4 stream cipher, and calculating a message authentication code (MAC). While this experiment is not a formal detailed verification of energy use, it does illustrate how encryption consumes extra energy.

The results of the second experiment are shown in Figure 5.11. 7,433 samples were

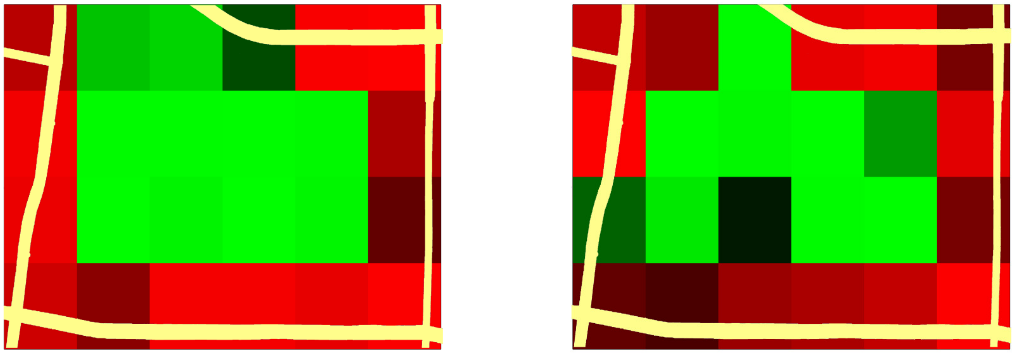


Figure 5.11: Experimental results of MDNSs accuracy when implemented on smart phones. Sound levels were sampled from the phone's microphone. Left panel: the original distribution of sound levels sampled from the environment, X. Right panel: the MDNS reconstructed distribution, A. Bright red squares are louder locations while bright green are quieter. Black represents the mean. Yellow denotes roads. One can conclude from the reconstructed distribution that the noisier locations correspond to those that contain the roads.

collected from 3 smart phones. From the figure it is clear that the louder locations are the perimeter locations, which appear as brighter red. The quieter locations inside the campus appear brighter green. Black represents the mean sound level. Color in the figure was determined from the slope of a linear regression with the reconstructed histogram of volume levels at each location, assuming histogram bins are one unit apart. Ideally, positive slopes indicates noisy environments. Samples were taken from the campus during rush hour between 4pm and 6pm. With approximately 7,300 samples, MDNSs were able to reconstruct the profile of sound in the environment.

5.3.2 KIPDA on Moteiv T-mote Invent Sensors

KIPDA was implemented on Moteiv T-mote Invent sensors. These motes use the TelosB architecture, have a bandwidth of 250 Kbps, and a CPU frequency up to

8MHz.

Implementation Setup

Three experiments were performed. The first experiment measured the times it took for the encrypt and decrypt primitives of the AES cipher. These times should reflect with results in the literature. The second experiment measured the times it took for each node to deplete its battery. Nodes using KIPDA should stay alive longer. The third experiment was similar to the second, except voltage information was taken from each node after 10 aggregations. With consistent current between all nodes and techniques, hop-by-hop aggregation with the AES cipher should draw more voltage.

For all three experiments one sensor (mote 0) was connected to the base station, a PC, which gave the device a constant supply of power. This mote communicated with the PC through a virtual serial port over USB. The same code base used in the TOSSIM simulations was used in the experiments with minor modifications, such as communicating with the PC. 10 sensors were used in total, and their topology is given in Figure 5.12.

Experiment One: This experiment measured the times to encrypt and decrypt for AES, RC5, SkipJack, Trivium, and TinyECC on a single mote. The cipher primitives (encrypt and decrypt) were performed 1000 times and averaged. Because timers on the mote did not provide adequate precision, the standard deviation is not reported, i.e., each individual measurement was below the minimum value of the timer.

Experiment Two: For the second experiment, I arranged the sensors in a tree topology inside a building with no partitions. With 9 sensors at full charge, I ran hop-by-hop *multiplicative* aggregation with AES encryption until each node's energy

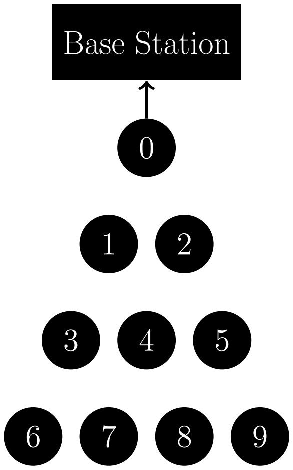


Figure 5.12: Topology used for the KIPDA implementations.

supply was insufficient to communicate. Since each node reported a unique prime number and multiplication aggregation was used, I could determine when nodes stopped communicating through prime factorization of the network-wide aggregate at the base station. The sensors were then recharged and the experiment was repeated with KIPDA sending 2 more bytes per message than AES. To determine when a node using KIPDA stopped communicating, each node after 10 aggregations sent its ID plus 100 to the base station. Nodes staggered their reporting between 10 aggregations, giving each a chance to tell the base station it was still alive. With MAX aggregation, and by the way the other nodes chose their sensitive data, a node's ID plus 100 was guaranteed to be the reported network-wide aggregate. If a node was down, the network-wide aggregate would be less than 100. Energy used by the two different aggregation functions (multiplicative aggregation for hop-by-hop encryption and MAX aggregation for KIPDA) is small enough that when compared to radio and encryption energy use it does not affect the experiment's outcome. For both tests, the maximum hop depth was set to 6, and an aggregation took place every 8 seconds.

Encryption Type	Time to encrypt (ms)	Time to decrypt (ms)
AES	1.847	2.162
RC5	1.314	1.273
SkipJack	0.466	0.453
Trivium	0.667	0.653
TinyECC	4,057.0	5,085.0

Table 5.4: Average time in microseconds to encrypt and decrypt a block or stream segment (size of which is given in Table 5.2) on a T-mote Invent sensor.

Experiment Three: This resembled Experiment 2, except that voltage information was recorded from every node every 10 aggregations.

Implementation Results

Experiment One: The results of Experiment 1 are shown in Table 5.4.

Experiment Two: The results are shown in Table 5.5 and differ on average by 1.1%, with the exception of nodes 2 and 3 which crashed during the KIPDA run. This small amount of difference between the two protocols was unexpected. It is possibly due to the variability of the underlying dissemination and collection protocols, temperature in the room, electronic noise in the environment, etc. To obtain better results, future work could re-design the underlying protocols, or have the sensors forward aggregates repeatedly after a certain amount of time, eliminating the need for the underlying dissemination and collection protocols.

Experiment Three: Assuming voltage drops are a proxy for the energy use in a battery, Figure 5.13 illustrates the voltage of mote number one, using both KIPDA and hop-by-hop aggregation with AES encryption. From the figure it is clear that encryption draws more voltage. Other motes show similar behavior.

	AES Encryption		KIPDA 18	
Node	Number of Aggregations	Time to failure (minutes)	Number of Aggregations	Time to failure (minutes)
1	17,872	2,327	17,937	2,336
2	18,335	2,388	16,338	2,128
3	18,153	2,364	13,639	1,776
4	18,864	2,457	18,850	2,455
5	18,371	2,392	18,351	2,390
6	17,738	2,310	17,532	2,283
7	17,080	2,224	17,903	2,332
8	18,166	2,366	18,194	2,370
9	17,541	2,284	17,945	2,337

Table 5.5: Comparison between AES hop-by-hop encryption and KIPDA with a message vector of 18 bytes. Motes running KIPDA should have a longer lifetime, even though 2 more bytes per packet are transmitted.

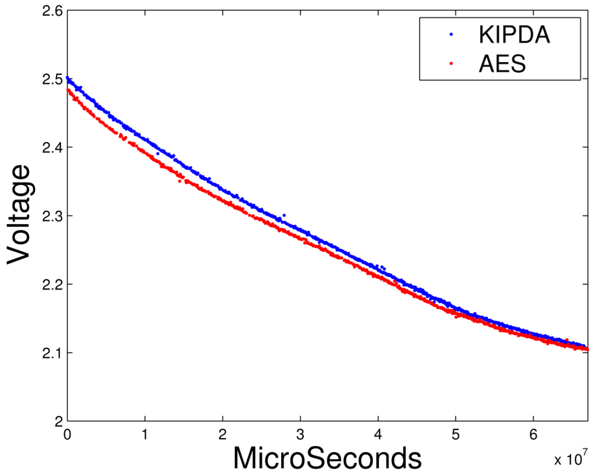


Figure 5.13: Comparison of voltage used between KIPDA with a message vector of 18 bytes, and hop-by-hop aggregation with the AES cipher that used 16 byte blocks. Comparisons took place on Moteiv T-mote Invent (TelosB architecture) sensors.

It should be noted that these 3 experiments do not use state-of-the-art techniques to measure energy usage in a device. For example, voltage alone does not give a sufficiently accurate estimate. A better technique would be to solder a resistor in parallel to the battery and use a multimeter to determine current. With voltage, current, and timing information, energy could then be measured accurately. This hardware solution is beyond the scope of my dissertation. Instead, I used three different low-technical experiments to measure energy, previously discussed. Two of these three experiments indicate that KIPDA uses less energy than hop-by-hop aggregation with AES encryption. The third was inconclusive. Future work can investigate if the third technique sent uniform sized packets in the lower network protocols, or if the amount of energy used in the lower networks protocols overshadow the energy used in the application layer.

5.4 Summary of Chapter

In this chapter, I presented two MATLAB simulations which illustrate the feasibility of using MDNSs to protect location privacy and data samples. TOSSIM simulations show that motes using KIPDA have extra available bandwidth because encryption is eliminated. While the privacy guarantee is not as strong as traditional encryption, it does protect in-network nodes from easily learning the sensitive information. An implementation of MDNSs on Android smart phones and an implementation of KIPDA on T-mote Invent sensors show that these techniques can be successfully deployed on physical devices. In each application, economies of scale were leveraged to achieve accurate reconstruction while protecting privacy and placing very low communication and computation overheads on the sensor nodes. The next chapter discusses how these applications can be improved, the trade-offs involved, and their strengths and limitations.

Chapter 6

Discussion

In this chapter, I provide examples of other possible applications, and discuss the strengths and limitations of MDNSs and KIPDA, along with their trade-offs between energy and privacy. I also present previous approaches that gave rise to MDNSs, and compare the three different techniques used to estimate energy usage of KIPDA: the analysis in Chapter 4, Section 5, the TOSSIM simulations in Chapter 5, Section 2, and an implementation on physical devices in Chapter 5, Section 3.2.

6.1 Possible Applications

Beyond radiation detection, perturbing continuous data, and determining noisy locations discussed in the previous chapter, there are several other possible application scenarios for both KIPDA and MDNSs. One such scenario includes public health, for example, collecting body temperatures and other data to detect disease outbreaks such as influenza. In privacy-preserving data mining, a data set of financial transactions could be perturbed with the MDNSs protocols, then released, allowing patterns of financial transactions to be discovered. Additionally, instead of detecting radiation with MDNSs, sensors could be located on vehicles to monitor air pollution, communicating with existing cellular phone networks. This could protect location privacy in the scenario presented by Zappi et al. [154]. Many eco-conscious drivers would be willing to participate in such a scheme, and participants with health conditions such as asthma could avoid certain parts of a city. Another scenario would include usage statistics of cable television channels, protecting the information of what a participant watches while allowing a station manager the ability to determine which shows to air. Additionally, MDNSs could be deployed in Africa to query human users through cell phones on their malaria [158] or HIV status, using their perturbed locations to identify possible outbreaks.

6.2 MDNSs

The previous chapters presented algorithms, evaluation metrics, simulations, and a prototype implementation for enhancing the privacy of participatory sensing applications with MDNSs. The approach is notable because it is computational and energy efficient and does not rely on encryption, key management, or a trusted base station. In this section, I summarize the tunable trade-offs among granularity (precision of data), accuracy (utility) and privacy, which were illustrated by the simulations and Android implementation. I then discuss the strengths and limitations of my current work, indicating areas for future extensions. Finally, I touch on some prior work that led to the current version of MDNSs.

6.2.1 Trade-offs

The balance among privacy, data granularity and reconstruction accuracy can be adjusted to meet the needs of a particular application as follows:

- **Data granularity:** Collecting information with finer granularity generally enhances the quality of the data, provided the number of participants scales accordingly. Given a constant number of participants, however, there is a trade-off between data granularity and reconstruction error. In some settings is may be preferable to report data with less precision (e.g., fewer locations, each covering a larger area) in exchange for higher accuracy of the reconstructed distributions.

- **Privacy:** Privacy increases with granularity. As the number of categories increases, it is more difficult for an adversary to guess the category that any individual mote has sensed [58].

- **Utility:** Utility suffers when the data have been altered or perturbed to a point that useful statistics can no longer be reconstructed. MDNSs manage this trade-off, maintaining usefulness while preserving privacy. With a fixed level of granularity, utility can be increased by adding more participants.

- **Energy:** While this may not be a tunable parameter in MDNS, it is worth noting the trade-offs between energy and privacy. Energy is conserved in MDNSs but the notion of privacy relaxed from 100% inability to determine sensitive data (such as in encryption) to k indistinguishability where the sensitive data cannot be determined from $k-1$ other data. This shows evidence of one of the main themes of this dissertation: more privacy requires more energy. In KIPDA, energy and privacy are actually tunable parameters.

6.2.2 Strengths

MDNSs using NSPMs enhance privacy in resource-constrained devices. The algorithms are efficient, all samples are guaranteed to be perturbed, utility metrics can be well-approximated regardless of the data distribution, they can be tuned using

DA to improve performance when the number of participants is limited, and they are the optimal Warner scheme perturbation matrix when the data distribution is not known *a priori*:

1. **Efficiency:** Because the algorithms at the nodes and the base station are so simple, the method is energy and computation efficient. The time complexity of the node protocol is $O(1)$ for each sensed value. M does not need to be stored or used in the perturbation process at the nodes. The base station's time complexity is the product of the number of categories for each dimension and the number of dimensions.

2. **All samples are guaranteed to be perturbed:** My algorithms use a perturbation matrix with zeros on all diagonal entries. If the matrix has non-zero values on the diagonal, a sample could, in principle, be reported with all of its original values. In some cases this would be viewed as a privacy breach even if it only occurred in 1 record out of a million [64].

3. **Utility is nearly independent of the prior data distribution:** The distribution of values in the environment is often not known before sensors are deployed. In Chapter 3, Section 3.3, I show that with a NSPM the effect of the underlying original distribution, X, on utility is small enough to safely assume it is independent. Other methods, however, require prior knowledge of the data distribution for computing an optimal perturbation matrix. For example, Hung and Du [89] argue that NSPMs (or any Warner scheme) are not the most optimal perturbation matrices for maximizing both privacy and utility. They use genetic algorithms to evolve an optimal perturbation matrix, taking privacy and utility metrics as components of the fitness function. Since their privacy metric assumes an underlying original distribution, X, the only way their scheme can evolve the best perturbation matrix is to know X.

4. **DA improves performance of NSPMs when the number of participants is limited:** Other perturbation matrices do not have straightforward implementations of this idea.

5. **NSPMs are the optimal Warner scheme when the prior data distribution is not known:** Figure 6.1 illustrates the optimality of different Warner scheme perturbation matrices when using 10 categories. The left panel of Figure 6.1 shows several data distributions (uniform, normal, random, uniform 50/50, exponential and spiked) with varying normalized Shannon entropies. The right panel plots privacy and utility metrics (y-axis) against different values of p (x-axis) in Equation (2.2). The spiked distribution is omitted from the right panel as its privacy is 1 for all values of p. Because utility is largely independent of the underlying distribution, the values appear as a single curve for all distributions. The underlying distribution, however, does affect privacy. The uniform distribution provides the best privacy (lower values are better) as it has the highest entropy.

The figure also illustrates the trade-offs between privacy and utility as p varies. As p approaches 0.1, utility increases asymptotically for all distributions. This occurs at 0.1 because if they are 10 categories, $p = 0.1$ implies that random values are being reported independent of what is sensed. Values closer to 0.1 also provide excellent privacy, but because of the asymptotic increase in utility (higher is worse) they are not viable parameter settings. Moving away from 0.1 improves utility symmetrically, but privacy does not degrade symmetrically. Because of this, $p = 0$ provides the same utility as $p = 0.2$, but with better privacy. Generally, lower p values will provide a better utility/privacy trade-off, however this breaks down when the underlying distribution does not have sufficient entropy, as is demonstrated by the exponential distribution.

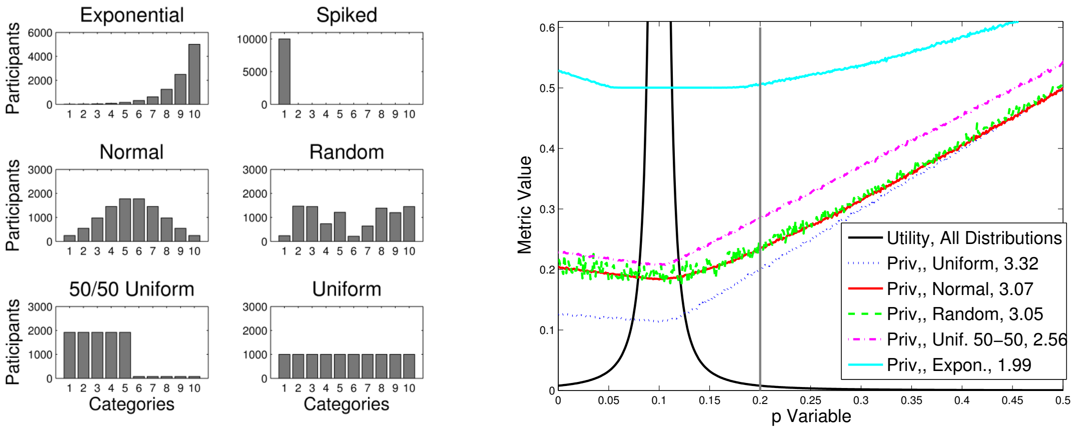


Figure 6.1: (Left) Six representative distributions with 10 categories and 10,000 participants used to compute data in left panel. (Right) Privacy and utility values using different values of p in Equation (2.2). Each curve represents the privacy value of a different underlying distribution, listed in the legend with its Shannon entropy. Utility is nearly the same for all 6 different distributions and is plotted once. I exclude the spiked distribution because it has a privacy value of 1 for all p.

In the cell phone implementation, because the data are perturbed, it is almost impossible for the collection server to determine a participant's true location. (A cell phone tower could potentially reveal the node's location, but the base station cannot determine individual locations from its own information.) Most, if not all, encryption methods must eventually trust the final recipient of the data. In contrast, MDNSs do not require such trust because the data are never "decrypted." My method also does not incur the extra computational and energy costs associated with encryption/decryption algorithms and the additional communication overhead required to transmit encrypted data. Finally, it does not require a key distribution and management system.

6.2.3 Limitations and Caveats

In the following, I discuss some limitations of the method and the experiments and how they might be addressed.

A sensor node might be captured by an adversary and report faulty data, either reporting the original sensed value or biasing the reported value in other ways. Similarly, if human inputs were solicited directly (e.g., please send us the name of one candidate you did not vote for, or which malaria symptoms you do not have), their (negative) answers might be biased [63]. This issue can be addressed if it is known how the negative answers are distributed, for example by adjusting the perturbation matrices used in reconstruction, M_δ, for each dimension δ, with the correct probabilities. A more extreme approach would authenticate the packets to ensure that they originate from a secure sensor or device.

If a mobile sensor is stationary, moving slowly or following a regular pattern, an adversary might be able to infer its location through long-term monitoring of the transmitted (negative) values. This would be especially important if an ID was required with the data. This threat can be mitigated if participants respond to a base station query only if their location has changed since the last query or to limit the amount of information sent to the base station if the participant's device determines more responses would decrease privacy.

For MDNSs that operate on continuous data with a small number of samples, repeated samples from the participants can be accumulated at the base station over a long period of time. For many participatory sensing applications, a long period of sensing is expected. However, as explained in the previous paragraph, nodes that continually sense the same information over long periods of time are prone to statistical attacks. Hence, this technique (continuous MDNSs) is more appropriate for privacy-preserving data mining. In addition, DA can improve accuracy of continuous

MDNSs data by using a lower base or radix for the samples. Also, if the first dimension (the most significant digit) contains a smaller range of categories, e.g. 0-5, this dimension can use fewer categories.

The reconstruction algorithm occasionally generates negative estimates for some categories. This statistical artifact arises when the expected contribution for a particular category exceeds the actual reported total for that category. As the number of samples increases, the number of negative estimates decreases, along with reconstruction error. If negative values appear in the reconstructed distribution, these can be mapped to zero. If the total number of participants must be consistent, then the negative amount can be deducted in equal amounts from the rest of the categories. For example, if I had a negative survey of ten categories and category one was reconstructed to negative nine, this value could be changed to zero and one value removed from each category between two and ten.

In the cell phone simulation each node reports directly to the base station. Routing in traditional wireless sensor networks is often organized as a tree with the base station at the root. Data aggregation in the network can improve efficiency but is challenging if privacy is important. This is an area of active investigation, for example, the work of Castelluccia et al. [24] and KIPDA.

The implementations indicated that duplicate packets need to be prevented. This could be solved with a unique user ID and packet ID numbers. The base station can authenticate participants' information, varying from an insecure but energy cheap technique such as accepting a unique ID, to a more secure and energy intensive technique such as signing the information with an asymmetric encryption key. Any authentication will increase data integrity, however, data integrity is left for future work.

In another limitation of MDNS, privacy protection depends on underlying distri-

Table 6.1: Cell phone radiation detection simulation for an example city such as Boston. Location was dimensional adjusted to 3 dimensions of 4, 4, and 2 categories each.

Locations adjusted to 3 dimensions of 4, 4 and 2 categories each					
	False	False	Acc. of True	Avg.	Avg.
Samples	Neg.	Pos.	Pos. % (Ratio)	Privacy	Utility
100,000	44	44	98.5 (449/456)	0.1013	1.5479e-05
200,000	7	5	100 (495/495)	0.0989	7.7398e-06
300,000	1	1	100 (499/499)	0.0980	5.1600e-06
400,000	0	0	100 (500/500)	0.0976	3.8700e-06
500,000	0	0	100 (500/500)	0.0972	3.0960e-06

butions with a sufficient amount of entropy. Take for example the following scenario that preserves location privacy. It is 3 am, there is only one night club in a city, and sensors monitor noise. A sensor that returns a low amount of noise can be determined to be at the night club. This was explored in Chapter 5, Section 3.3, where the spiked distribution (low entropy) from Figure 6.1 (left) gave a privacy metric value of one, the worst privacy.

Another example of trade-offs occurs in the cell phone radiation detection simulation where the k value of k-indistinguishability with DA using 4 dimensions is 6. If 32 locations are used, this increases utility, however, the granularity of a location also increases. If the 32 locations use DA with 3 dimensions of 4, 4, and 2 categories each, a higher k value of 9 is obtained. These parameters are realistic for a city such as Boston, where each location would represent 1.5 square miles. Simulations similar to those presented in Chapter 5, Section 1.1 were performed for this scenario and the results are shown in Table 6.1. It takes twice as many samples to obtain 0 false negatives and false positives and 100% determination accuracy than the simulations presented in Chapter 5, Section 1.1, however, this simulation increases k from 6 to 9.

MDNSs with DA do not perform well with a high number of categories. For example, if DA sets all dimensions as two categories except for the last dimension, which is set as three categories (this provides the best utility), I could only reconstruct music usage statistics [1] with good enough utility to obtain in a reasonable order the top 100 listened to artists when the overall number of categories (artists) was less than 3,000.

6.2.4 Previous Approaches

The idea of MDNSs stemmed from spatial negative surveys where reconstruction is based on the assumption that neighboring nodes have similar values. This assumption allows accurate reconstruction of the original environment from the perturbed readings, as illustrated in Figure 6.2 for a gradient temperature map of a 1000x1000 grid of sensors. Each sensor perturbs its temperature reading once and sends the perturbed datum back to the base station. The base station then takes a sliding window over the environment centered on the node it is estimating. It assigns to this node the value that appears least often in the window. The caveat with this technique is that the base station could infer the sensitive value for each node. However, this technique gave rise to MDNSs.

It was originally thought that the reconstruction problem in MDNSs was NP-complete. The first devised solution was based on the box packing problem. The next solution was based on an extension to the SDNS reconstruction equation and is similar to the *inclusion exclusion principle* [15] (also known as the sieve principle), yet, it also had exponential complexity. I then used the Kronecker technique to further improve the running time. Finally, matrix memoization was created to cache calculations back into the disguised distribution's matrix to achieve reasonable polynomial time complexity.

[1] Data set used was from the Yahoo KDD Cup Challenge [49].

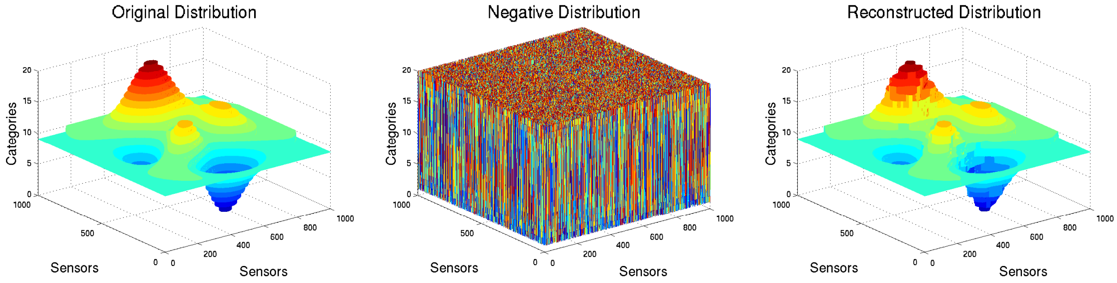


Figure 6.2: A spatial negative survey. From left to right: The original environment, the negated readings, the reconstructed environment. The sensors are evenly spaced on a one hundred by one hundred grid. The z^{th} dimension is the sensor readings. Each sensor sends back only one reading.

6.3 KIPDA

In this section, I summarize the adjustable trade-offs between privacy, efficiency, and accuracy which were illustrated by the analysis in Chapter 4, Section 5.1, the simulations in Chapter 5, Section 2, and the implementations in Chapter 5, Section 3.2; I later discuss the accuracy of estimating energy usage between these three sections and I present the strengths and limitations of KIPDA, indicating areas for future extensions.

6.3.1 Trade-offs

There exists a spectrum of privacy-preservation versus energy usage. On one extreme exists maximum privacy, for example public/private key encryption. Unfortunately methods such as these consume a great deal of energy and other computing resources. On the other end of the spectrum is the case of no privacy. This uses the least amount of energy but provides no data protection. I examine the area in between and use the following characteristics to aid in explaining this spectrum:

1. **Privacy:** Different notions of privacy ensure that data are either, 1) not revealed, 2) partially revealed, or are 3) indistinguishable from other data. Ideally the sensitive value d_i of node i should not be known to any other nodes j in the network, or outside observers listening to radio communications. However, this notion can sometimes be relaxed to conserve energy. For example, KIPDA provides k-indistinguishability that hides sensitive data among $k-1$ items of camouflage data. CDA schemes should also be robust enough to tolerate collusion among several nodes, at least to some extent, such as with KIPDA.

2. **Efficiency:** Data aggregation is able to reduce the number of messages transmitted within the sensor network, thus reducing bandwidth and energy use. However, additional overhead is introduced to protect privacy. If the energy cost of a CDA scheme is greater than the benefit of data aggregation, it accomplishes nothing. A good CDA scheme should have low overhead. Bandwidth, energy consumption, and delay are three important metrics to measure the protocol efficiency. KIPDA reduces the overhead associated with encryption for a modest increase in bandwidth. For optimal energy use, the increase in bandwidth should be less than the overhead of encryption.

3. **Accuracy or Utility:** Accuracy is an important characteristic of any CDA scheme, as a low or inaccurate aggregation result may affect the decisions made from these results. Accuracy and utility are usually sacrificed to achieve privacy and efficiency. However, KIPDA achieves 100% accuracy in an efficient manner, at the cost of a different notion of privacy.

Trade-offs among privacy, efficiency, and accuracy are complicated and must be tailored to different applications. These considerations highlight the importance of choosing the proper parameters for different applications.

6.3.2 Strengths

KIPDA keeps MAX/MIN information protected from other in-network nodes. To determine the sensitive information several nodes have to collude. Thus, KIPDA provides a level of privacy in an honest-but-curious network, and a higher level of privacy to outside observers. It conserves overall energy usage in a network, but energy use can be increased to provide more protection. Additionally, KIPDA provides fast aggregation, which can conserve energy if nodes use an LPL scheme.

6.3.3 Limitations and Caveats

If values are the same or even correlated after every aggregation epoch, the sets GSS and R^i can be changed over time. This would also help against statistical attacks that monitor message vectors. One solution involves the base station securely assigning R^i and P^i to every i. Because this consumes extra energy, it needs to occur occasionally. One way to disseminate this information is to establish session keys. Session keys can be distributed with an asymmetric (public/private key) encryption method such as TinyECC. Once the session keys are distributed, cheaper symmetric encryption can be used to disseminate R^i and P^i.

Ideally, instead of continually distributing P^i and R^i, a symmetric key could be used in a random number generator that generates just enough information for R^i and P^i, but not enough so that i can determine R^j or P^j for any j. In this sense, the problem is not completely solved. Hopefully, this research will inspire a new technique that can generate and keep secret a set of numbers that are partially shared among nodes. This KIPDA random number generator is left for future work.

One possible attack is for an adversary to control the external property of a sensor and examine the message vector to find the sensor's restricted slots. For example, she might put an ice block on top of a sensor so that the measured temperature is 0 degrees. Then, she can guess the restricted slots by looking at values less than 0 (for MAX aggregation). A possible solution, assuming the adversary cannot examine the internal state of a sensor, is to have the sets R^i and P^i change every aggregation epoch. Another solution is to have nodes vote out an aggregate that does not meet the typical distribution in the environment. Another possibility is to tighten the range camouflage values are drawn from to a range that is closer to the sensed value, or, assume the environment will have a high enough entropy of sensed values.

KIPDA works best for data that have a small range, i.e., use a small number of bits. Larger ranges such as 16 bits of data or more could use an excessive amount of energy for some applications. There is a relatively small number of messages that can be sent to maintain energy savings compared to the Trivium, RC5, and SkipJack ciphers which the threshold ratio predicts as 7, 10, and 11 bytes respectively. Energy use in KIPDA is compared to hop-by-hop encryption aggregation, however, hop-by-hop encryption does not secure information against other nodes. If security against in-network nodes is truly desired, and the energy is available, then the message vector size in KIPDA could be increased. Additionally, the same argument applies if fast aggregate responses are necessary. In this case, however, the underlying dissemination and collection protocol would need to be optimized for speed and energy. This is left for future work.

It is difficult in MAX/MIN aggregation to fill the restricted set, if the theoretical minimum or maximum values are sensed. One way to solve this is to assume that an adversary does not know these theoretical values, and to increase the range of values in the message vectors beyond them.

6.3.4 Comparisons of Energy Analysis, Simulations, and Implementations

This section compares the energy estimates between KIPDA and hop-by-hop aggregation with various forms of encryption from the analysis given in Chapter 4, Section 5.1, the TOSSIM simulations given in Chapter 5, Section 2, and the implementation on T-Mote Invent sensors given in Chapter 5, Section 3.2.

Since RC5 on the MicaZ architecture was used in both the analysis in Chapter 4, Section 5.1 and the TOSSIM simulations in Chapter 5, Section 2, they can be compared. From Table 4.7 in the analysis, the average RC5 primitive took 7.03 milliseconds, using 181.5 microjoules. From Table 5.3 in the TOSSIM simulation, the average RC5 primitive took 0.2483 milliseconds and used 5.63 microjoules. The analysis in Chapter 4, Section 5.1 overestimated encryption costs by 3,124% for energy and 2,821% for time. When comparing RC5 on the TelosB architecture, the analysis in Chapter 4, Section 5.1 overestimated the time to encrypt and decrypt by 401% compared to the time reported by the implementation on a TelosB device in Chapter 5, Section 3.2. Energy results for the encryption primitives of RC5 were not obtained from the TelosB device, but could be obtained for future work.

When comparing encryption and decryption primitive times in the literature to the TOSSIM simulations, RC5 and SkipJack come close to what is reported by Singh and Muthukkumarasamy [136]. RC5 (the average between encrypt and decrypt primitives) deviates by 1.8 μs or 0.7%, while SkipJack deviates by 6 μs or 1.7%. The difference could be explained by the different architectures. Singh and Muthukkumarasamy [136] used Mica2 physical devices, while I used a MicaZ emulator. Because the results from the simulator agree with what is in the literature, it appears that the analysis in Chapter 4, Section 5.1 needs re-evaluation. However, this is why I implemented KIPDA in simulators and on physical devices — to determine the correctness of the analysis.

In summary, the original analysis, based on the estimates taken from the literature, overestimated the energy and time costs of encryption. Also, based from the literature, the energy costs determined from the TOSSIM simulations appear close to the actual energy and time costs observed on physical Mica2 devices.

Chapter 7

Related Work[1]

I begin with a general literature review, and then compare separately literature that is specific to MDNSs and KIPDA.

Privacy-preserving algorithms have been developed for data mining [64, 90, 95, 127], data aggregation [26, 65, 71, 73, 83, 85], and other applications [108, 138]. There are four main classes of solutions: perturbation, k-anonymity, secure multi-party computation, and homomorphic encryption.

In data mining, data values are typically hidden by perturbing individual data or query results [64, 90, 95]. To obtain accurate results, these methods typically assume that the distribution of data/noise is known ahead of time. However, as shown by Kargupta et al. [95] and Huang et al. [90], certain types of data perturbation might not preserve privacy well. MDNSs on continuous data is another perturbation approach for data mining which does not assume the distribution of data is known ahead of time.

[1] Related work was left till later in the dissertation to give the reader enough information to compare my approaches with existing techniques.

K-anonymization [5, 113, 133, 139] in WSNs makes a participant indistinguishable from $k-1$ other nearby participants. It was originally designed for privacy-preserving data mining, but in participatory sensing applications individual participants can sense and share their own data. Thus, there is limited potential to mix participants' data with others' data as required for *k*-anonymity. MDNSs protect data before it leaves the individual's device, with more potential to mix participants' data.

Secure multi-party computation (SMC) [38, 81, 88] methods specify a joint computation among a set of involved peers. This is problematic in a participatory sensing setting, because of high communication or computation overhead when the participant population is large, and some participants may not trust their peers.

There is a growing body of work developing techniques for aggregating data that have been encrypted using homomorphic functions [26, 56, 73], which allows a user to calculate some aggregate values (e.g., product, or summation, or both) using the encrypted values. However, encryption, such as those based on the Domingo-Ferrer's [46] symmetric key technique, is energy-intensive, which can limit its applicability to resource-constrained devices. Additionally, only summation and/or multiplication aggregate functions can be computed using these approaches, and for some techniques, in order to interpret the final aggregation result, a server needs to know which nodes reported data to know which encryption keys to use [24, 26], which is not always desirable.

A slightly different approach from the previous four is taken in SMART [83], which slices individual data, sending the slices through the aggregation network, and reassembling the data pieces later. However, SMART requires the availability and trust of neighboring peers, which may not always be available.

7.1 MDNSs

Randomized response techniques (RRTs) have been used to obtain answers to sensitive questions when respondents might be reluctant to answer truthfully [89, 147]. While originally designed for dichotomous populations, they were generalized by Abul-Ela et al. [2] for polychotomous responses by using several different samples. Bourke and Dalenius [18, 19] suggest a different scheme that uses a single sample. While RRTs and my method both strive to protect privacy in surveys, RRTs use the random device to choose among several *questions*, where at least one is sensitive; while in negative surveys the perturbation matrix guarantees no one answers with the sensitive question/data. The subjects in RRTs that are asked the sensitive questions must ultimately trust the information collector, while negative surveys never require participants to respond honestly to any sensitive questions.

The most optimal perturbation matrix for RRTs has received some research. Agrawal and Haritsa initially studied the problem [11]; however, they only focused on symmetric matrices and used only accuracy (utility) for comparison. Huang and Du achieved better results by using a multi-objective optimization technique to study various matrices with measures of both privacy and utility [89]. However, while they state their method can be used for multi-dimensional data, they only studied one dimension. They argued that NSPMs, and ultimately all schemes based on the Warner matrix are inferior, while Chapter 6, Section 2.2 suggests they have an advantage in resource-constrained devices and WSNs.

A common practice for privacy applications is to remove users' identifiers; however, multiple attributes of the records or external data can often be combined to uniquely identify individuals [139]. Previous privacy-preserving proposals for multi-dimensional data were designed for databases and data mining, where a database server stores records of multiple users. In this situation, the server is able to infer users' data from others [5, 113, 133, 139] and alter or change the responses to protect privacy; however, this assumption is not true in participatory sensing applications, where individual users have only their own sensed information to report. MDNSs are a reverse of this scenario, where the client alters the response to protect herself from the server.

In participatory sensing applications, data points are often tagged with location information, and a rich set of location-based privacy and anonymity rules have been developed for this situation [112, 126]. These schemes, however, typically hide or perturb single-dimensional continuous location information. For multi-dimensional data, privacy-preservation involves trade-offs among accuracy (or information completeness), computational complexity, and the level of anonymity. Aggarwal et al.[5] has shown the curse of high dimensionality for k-anonymization in data mining, even if k=2. MDNSs are able to handle about 3,000 categories before the curse of dimensionality affects accuracy.

Dora et al. [48] present a similar idea that hides or lies about true information. This idea works on delay tolerant networks with the store-carry-and-forward principle by perturbing categorical information. Their work differs from mine because they perturb a bit vector, a user interest profile, so that it cannot be linked to a user. As a result of this perturbation, a node can hide its interest in a category so that less messages of that category are stored, carried, and forwarded; or it can lie about its interest in a category so that more messages are stored, carried, and forwarded. MDNSs perturb the data that are sensed and communicated, instead of sending more or less information.

Dwork et al. [54] introduced the term *pan-private* in the context of streaming algorithms which can protect the state of information inside a node. This is useful for protecting against node capture attacks that examine internal data. However, it assumes a secure stream as a precondition of the algorithm. In contrast, the work reported in Chapter 3 protects the stream of information in transit. Pan-private algorithms, however, are preferable for complex aggregates such as the t-incidence items, the t-cropped mean, and the fraction of k-heavy hitters [54].

Differential privacy [52, 53, 149] aims to provide the maximal accuracy of responses for users querying a statistical database, while minimizing the ability of these users to identify the records in the database. Differential privacy assumes that a trusted server handles and responds to the queries, while negative surveys, on the other hand, do not assume that the server is trustworthy.

Negative surveys [59] are closely related to *negative databases* (NDBs) [60, 61, 62], which are an alternative representation of information that stores the set complement of data instead of the actual data. Both negative surveys and NDBs store or report data in this way, yet NDBs differ because the information is stored in a compressed form that adds additional security. It is provably NP-complete by a reduction to 3SAT, to try and extract the positive database from the NDB. Negative surveys, unlike NDBs, do not store the entire set of the strings representing the data complement.

Silence in communication [157] is similar in spirit to the negative representations of information, yet it does not attempt to provide anonymity. Instead, the protocols attempt to reduce communication overhead. This is accomplished via start and stop tokens, where the amount of time between the tokens represents the transmitted data. This method has the advantage of low communication overhead, although transmission bandwidth may suffer. Currently, a complete protocol based on this unique idea does not exist, and would seem problematic with current carrier sense multiple access with collision avoidance (CSMA/CA) [140] protocols.

Gaussian negative surveys (GNSs) reduce the number of participants needed for accurate negative survey reconstruction. Xie et al. [150] propose a special perturbation matrix where the cells in each column of the matrix assume Gaussian distribution values with the mean centered over the original category, which is represented as zero. With location data, this perturbs an individual's location a Gaussian random distance away from the original location. This special perturbation matrix eliminates the need for reconstruction at the base station. However, GNSs with location data do not protect privacy as well as traditional negative surveys. The privacy guarantee of an individual participant depends on the variance of the Gaussian distributions in the perturbation matrix. This variance must be small enough to maintain an acceptable level of utility and number of participants, however, smaller values do not perturb a location a sufficient amount of distance. This may make it easier for an adversary to determine the general location of an individual participant. It is not until the variance is increased to cover more than the entire column of the perturbation matrix that GNSs approach the same privacy guarantee as traditional negative surveys.

Quercia et al. [130] propose a randomized response technique similar to MDNSs. Instead of perturbing a location to a different location, each location is perturbed to a yes or no bit with a probability that includes whether a participant is at that location. For each sample a bit vector whose size depends on the number of locations, $O(\text{number of locations})$, is transmitted, while MDNSs transmit a smaller vector, $O(\log(\text{number of locations}))$.

MDNSs could use a scheme similar to the negative quad tree proposed by Horey et al. [84] where locations are nested inside other locations. Each level would represent a dimension in MDNSs. In their scheme, levels were recursively divided into 4 categories. MDNSs can use an arbitrary and varying number of levels and categories. However, using a negative quad tree or similar approach could increase the total area coverage in the cell phone simulation from a city to a metropolitan area.

7.2 KIPDA

Data aggregation without privacy achieves bandwidth and energy efficiency in resource-limited WSNs [109]. Previous work [1, 33, 44, 91, 92, 104, 137, 141] addresses data aggregation in various application scenarios with the assumption that all sensors are working in trusted and friendly environments. However, sensor networks are likely to be deployed in an untrusted environment, where links can be eavesdropped and messages can be altered. LeMay et al. [102] summarize the functional characteristic of wireless metering sensors and categorize attacks, where both privacy and security are concerns in the given scenarios. Previous work [29, 128, 151] investigates secure data aggregation against adversaries who try to tamper with the intermediate aggregation result. CDA is also closely related to and has been studied in the data mining domain [9, 90, 95] and peer-to-peer network applications [88].

The most secure data aggregation methods use either a symmetric or an asymmetric key approach. Symmetric keys, which use less resources, are similar to KIPDA which uses a global symmetric key approach, but differs in the sense that each node has a random part of the global key. The key is, in essence the set GSS. Only by capturing enough nodes will an adversary determine the correct global key in its entirety.

KIPDA differs from hop-by-hop and end-to-end encryption aggregation. Information is kept confidential from other nodes, which is not the case with traditional hop-by-hop encryption. End-to-end encryption is expensive since most implementations involve public/private key encryption. Other less expensive techniques that use symmetric keys such as the one presented by Castelluccia et al. [26] require the base station to know every node that participated in the aggregation.

Although the concept of camouflage has not to the best of my knowledge been applied to data aggregation, it has been applied to routing methods [45, 76]. Conner et al. [35] use decoy sinks and perturb network traffic to protect the location of the real sink.

k-Indistinguishability is closely related to k-anonymity [5, 6, 13, 139] which is designed to prohibit linking attacks. In a linking attack, an adversary matches auxiliary information with public or broad-casted information to determine the identity of one or more individuals. In contrast, KIPDA ensures the indistinguishability of the data itself instead of the identity of individuals or the source of the data.

Girao et al. [73] developed end-to-end encryption for the average and movement detection functions using Domingo-Ferrer's privacy homomorphism. Privacy homomorphism, however, was shown by Rivest et al. [131] to be insecure against ciphertext only attacks, if a comparison operation is supported. As a response, Acharya et al. [3] apply a type of privacy homomorphism developed by Agrawal et al. [8] called order preserving encryption scheme (OPES), to WSNs. In this scheme, nodes map their plaintext measurements to a set of ciphered values which preserves the order of the measurements. However, this scheme cannot prevent in-network nodes from learning private data if all sensors use the same set of mapping functions.

Zhang et al. [156] provide an aggregation of histograms where values are hashed. However, they can only produce approximates of the maximum or minimum value in the network, which will not work with WSNs that need accurate information.

Ertaul et al. [55] proposes an alternative scheme to OPES, where the maximum and minimum functions are computed using addition in a secure homomorphic encryption scheme. They encrypt messages of zero and z, where z is non-zero. This encryption does not have to take place at the sensor nodes as long as the results are stored in the nodes. Their scheme is secure against eavesdropping, and node collusion. However, the communication cost does not scale well to large sensed values, even sensed values that use 8 bits. This is because n encrypted messages need to

	MAX/MIN Ability	Accurate	Efficient	Level 1 Privacy	Level 2 Privacy
Girao et al. [73]	No	Yes	Yes	Yes	Yes
Ertual et al. [55]	Yes	Yes	No	Yes	Yes
Zhang et al. [156]	Yes	No	Yes	Yes	Yes
Yao et al. [153]	Yes	Yes	Yes	Yes	No
Groat et al. [79] (KIPDA)	Yes	Yes	Yes	Indistinguishablity	partial resilience

Table 7.1: Comparison of different secure data aggregation technologies.

be sent, where n is the largest sensed value. For sensed values with 16 bits, this is over 65,000 messages per node, per aggregation. Additionally, each message needs enough bits to protect against statistical attacks.

Yao et al. [153] provide another secure MIN/MAX scheme. Their scheme uses more energy than hop-by-hop aggregation with RC5 encryption. Additionally, it is insecure to eavesdroppers or colluding nodes if the global key is compromised, which can be accomplished by capturing one node. With the global key, an adversary can encrypt ranged HMAC messages such as $[d_{adversary}, d_{top}]$ or $[d_{bottom}, d_{adversary}]$, where $d_{adversary}$ is chosen by the adversary and d_{top} and d_{bottom} are the max and min network wide values. She could then test against captured non-range HMAC messages which contain the sensitive value. The value $d_{adversary}$ can be lowered or raised iteratively until the intersection of the captured message and the adversary ranged message are no longer non null, which will reveal the sensitive information.

Chapter 8

Future Work and Conclusion

8.1 Future Work

In this section I outline future work for MDNSs and KIPDA, in addition to general future work related to this dissertation.

8.1.1 MDNSs

A major assumption of negative surveys [85] is that the data from different sensor nodes are not correlated. I assume that each measurement made by a sensor node is independent of the other sensor nodes. However, this may not be the case in all applications. Two sensor nodes placed in close geographic proximity may sense correlated temperature values. Therefore, by knowing the locations of these nodes, more information could be gained from a single negative reply. Similarly, data reported by a sensor node may be correlated with past data. This correlation could be used to my advantage to estimate the individual data values and perhaps aggregate the results. Future work will examine the effects of correlation in the underlying data in single-dimensional and multi-dimensional negative surveys, quantifying the amount of privacy lost, and finding a solution that perhaps aggregates the correlated data efficiently.

Future work on MDNSs can examine non NSPMs that make the distribution of disguised data uniform, with the constraint of zeros on the diagonals. I propose to study whether the reconstruction error decreases if I choose the negative categories with a non-uniform probability such that the distribution of negative histograms is uniform. Preliminary results show that it is sometimes impossible to achieve complete uniformity since it involves inverting a non-invertible matrix. Yet, if the probability matrix were chosen to maximize uniformity of the negative histogram, this might have a beneficial effect on the reconstruction error. The perturbation matrix could then be transmitted instead of the negative data, especially if it is sparse.

Future work could examine the limits of DA on real-world data sets with large numbers of categories. Although histograms are useful, other aggregates could be explored. Additionally, a privacy metric for MDNSs based on indistinguishability adjusted by the underlying distribution could be useful. Furthermore, I could devise more complicated and accurate methods for threat determination in the radiation detection simulation. Another possible idea is to report the histogram, but randomize the bins in the histogram, perhaps at each hop level.

8.1.2 KIPDA

Energy usage of KIPDA could be compared to other schemes such as OPES, or the schemes presented by Ertaul's et al. [56] and Yao et al. [153]. Additionally, energy comparisons with homomorphic encryption which provides a good level of privacy protection would be beneficial if KIPDA could be extended to the summation and multiplication aggregation functions.

KIPDA could be modified to use variable sizes for the sets I, GSS, and R^i. Further future work could investigate how to distribute efficiently the sets R^i (or U^i) and P^i to each node. These two suggestions would help thwart adversaries monitoring repeated aggregations.

Results from Meulenaer et al. [42] conclude that energy is wasted in an LPL scheme when the radio has to wait for encryption. LPL could be implemented in TinyOS, TOSSIM, and PowerTOSSIM-Z to verify if KIPDA has an advantage over encryption aggregation schemes.

Additional future work could implement or use existing energy-conserving dissemination and collection protocols and convert KIPDA to report other order statistics such as the mode, mean, median, or k-heavy hitters.

8.1.3 General Future Work

Future work in CDA could create a method that will not only securely report the MAX or MIN aggregate, but securely report which node generated the value. Fast data aggregation is starting to appear in the literature [129] which could benefit from privacy-protection schemes similar to KIPDA. Additionally, future aggregation work could devise a homomorphic compression scheme, where the aggregation function is performed on the compressed data.

The computational power in WSNs may eventually increase to allow more complicated algorithms to run on them, such as those from machine learning. These collaborative machine learning algorithms will depend on the amount of energy available to determine a sensor's lifetime, and ultimately their ability to learn. While the complexity for machine learning algorithms currently depends on their input, they may eventually depend on the energy available [1].

[1] Idea from Terran Lane.

Future work could incorporate data integrity into both MDNSs and KIPDA by authenticating data and determining whether it has been altered. Also, indistinguishability and negative representations of data could be applied to different fields such as cloud computing or privacy preserving data mining; future work will continue to look for new applications that can incorporate these ideas.

8.2 Concluding Remarks

MDNSs provide a more efficient and robust way to protect individual information while providing the utility to mine group information. Information such as physical locations, driving speeds, or medical data can have devastating effects if intercepted by adversarial parties. MDNSs perturb data for participatory sensing applications, providing high levels of privacy. The privacy preservation problem addressed here is challenging, because (1) users may not trust the information collection server, and (2) embedded or sensor devices may have limited resources. Therefore, I do not rely on standard encryption schemes or key distribution and management. MDNSs scale well because the communication and computational overhead is low for the sensor nodes, especially when compared to expensive encryption and key management schemes. An advantage of my work is that privacy and accuracy can be managed by simply tuning parameters of the protocols, as evident from the simulations and implementations in Chapter 5. If the base station receives enough information, an aggregate distribution in multiple dimensions can be reconstructed efficiently and accurately. The DA technique produces less reconstruction error with the same number of participants. Continuous MDNSs have implications for privacy-preserving data mining where there is minimal research on reconstructing perturbed multi-dimensional categorical data.

KIPDA is the first work I am aware of that provides indistinguishability to CDA. It saves energy and time even though more messages are transmitted over the radio.

While encryption provides a stronger level of privacy, I have shown in Chapter 4, Section 5, and Chapter 5, Sections 2 and 3.2 that it is more energy efficient to slightly increase radio usage with decoys than to use conventional methods of hop-by-hop encryption. I have also shown how WSNs can protect confidentiality by hiding sensitive values in plaintext along with decoy values. By allowing the sensitive values to be in plaintext, aggregation can take place efficiently, which would otherwise be difficult. Confidentiality is achieved through k-indistinguishability with the sensitive aggregates hidden among $k-1$ other values. Dividing a message vector into different subsets (R^i, P^i, and U^i) provides the capability to camouflage message vectors with restricted and unrestricted decoys. A semi shared global key, GSS, creates resistance to node collusion and capture attacks.

Trade-offs exist between energy and privacy. This dissertation has shown that by relaxing the notion of privacy to k-indistinguishability, energy can be conserved. In KIPDA, sensitive data in a message vector are indistinguishable from some of the other members of the message vector depending on the number of nodes colluding, or radio messages intercepted. MDNSs also provide k-indistinguishability, where location information is indistinguishable from k other locations where k depends on whether or not an adversary has knowledge of the underlying distribution. These algorithms are appropriate for resource-constrained devices as they reduce energy usage and protect what they are sensing.

Appendices

Appendix A

Publications from this Dissertation

- M. Groat, B. Edwards, J. Horey, W. He, S. Forrest. “Applications and Analysis of Multi-dimensional Negative Surveys in Participatory Sensing Applications.” In submission to PMCJ 2012.

- J. Horey, S. Forrest, M. Groat. “Reconstructing Spatial Distributions from Anonymized Locations.” In ICDE Workshop on Secure Data Management on Smartphones and Mobiles, April 2012, Washington DC, USA.

- M. Groat, B. Edwards, J. Horey, W. He, S. Forrest. Enhancing Privacy in “Participatory Sensing Applications with Multi-dimensional Data.” In Proceedings of the Tenth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom ’12), March 2012, pp. 144-152, Lugano, Switzerland.

- M. Groat, W. He, S. Forrest. “KIPDA: k-Indistinguishable Privacy-preserving Data Aggregation in Wireless Sensor Networks.” In Proceedings of the Thirtieth Annual IEEE International Conference on Computer Communications (InfoCom ’11), April 2011, pp. 2024-2032, Shanghai, China.

- J. Horey, M. M. Groat, F. Esponda, S. Forrest. “Anonymous Data Collection in Sensor Networks.” In Proceedings of the Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services (Mobiquitous ’07), August 2007, pp. 1-8, Philadelphia, PA, USA.

References

[1] T. Abdelzaher, T. He, and J. Stankovic, "Feedback control of data aggregation in sensor networks," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, Dec. 2004.

[2] A. Abul-Ela, B. Greenberg, and D. Horvitz, "A multiproportions randomized response model," *Journal of the American Statistical Association*, vol. 62, pp. 990–1008, Sep. 1967.

[3] M. Acharya, J. Girao, and D. Westhoff, "Secure comparison of encrypted data in wireless sensor networks," in *Proceedings of the 3rd International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, Washington, DC, USA, 2005, pp. 47–53.

[4] J. Agajo and A. Theophilus, "Using wireless sensor networks for industrial monitoring," in *IET International Conference on Wireless Sensor Network*, Nov. 2010, pp. 113–121.

[5] C. C. Aggarwal, "On k-anonymity and the curse of dimensionality," in *Proceedings of the 31st International Conference on Very Large Data Bases*, Trondheim, Norway, Aug.–Sep. 2005, pp. 901–909.

[6] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu, "Approximation algorithms for k-anonymity," *Journal of Privacy Technology*, Nov. 2005.

[7] D. Agrawal and C. C. Aggarwal, "On the design and quantification of privacy preserving data mining algorithms," in *Proceedings of the 20th symposium on principles of database systems*, Santa Barbara, CA, USA, May 2001, pp. 247–255.

[8] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, “Order preserving encryption for numeric data,” in *Proceedings of the 2004 ACM SIGMOD International Conference on management of data*, Paris, France, 2004, pp. 563–574.

[9] R. Agrawal and R. Srikant, “Privacy-preserving data mining,” in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, TX, USA, Jun. 2000, pp. 439–450.

[10] R. Agrawal, R. Srikant, and D. Thomas, “Privacy preserving OLAP,” in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, Baltimore, MD, USA, Jun. 2005, pp. 251–262.

[11] S. Agrawal and J. R. Haritsa, “A framework for high-accuracy privacy-preserving mining,” in *Proceedings of the 21st International Conference on Data Engineering*, Tokyo, Japan, Apr. 2005, pp. 193–204.

[12] S. Aslam, F. Farooq, and S. Sarwar, “Power consumption in wireless sensor networks,” in *Proceedings of the 7th International Conference on Frontiers of Information Technology*, Abbottabad, Pakistan, 2009, pp. 14:1–14:9.

[13] M. Atzori, F. Bonchi, F. Giannotti, and D. Pedreschi, “k-Anonymous patterns,” in *Proceedings of the Principles and Practice of Knowledge Discovery in Databases*, 2005, pp. 10–21.

[14] A. Awang and M. Suhaimi, “Rimbamon: A forest monitoring system using wireless sensor networks,” in *International Conference on Intelligent and Advanced Systems.*, Kuala Lumpur, Malaysia, Nov. 2007, pp. 1101–1106.

[15] V. K. Balakrishnan, *Theory and Problems of Combinatorics*. Schaum’s Outline Series, McGraw-Hill, 1995.

[16] N. Bandirmali, I. Erturk, and C. Ceken, “Securing data transfer in delay-sensitive and energy-aware WSNs using the scalable encryption algorithm,” in *4th International Symposium on Wireless Pervasive Computing*, Feb. 2009, pp. 1–6.

[17] A. Bertaud and S. Malpezzi, “The spatial distribution of population in 48 world cities: Implications for economies in transition,” 2003, Unpublished manuscript.

[18] P. D. Bourke and T. Dalenius, “Multi-proportions randomized response using a single sample,” University of Stockholm, Institute of Statistics, Tech. Rep., 1973.

[19] ——, “Some new ideas in the realm of randomized inquiries,” *International Statistics Review*, vol. 44, pp. 219–221, 1976.

[20] M. Buettner, R. Prasad, A. Sample, D. Yeager, B. Greenstein, J. R. Smith, and D. Wetherall, "RFID sensor networks with the Intel WISP," in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, Raleigh, NC, USA, 2008, pp. 393–394.

[21] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing," in *World Sensor Web Workshop, ACM Sensys*, Boulder, CO, USA, Oct. 2006.

[22] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. A. Peterson, "People-centric urban sensing," in *Proceedings of the 2nd Annual International Workshop on Wireless Internet*, Boston, MA, USA, Aug. 2006, pp. 2–5.

[23] S. Camtepe and B. Yener, "Combinatorial design of key distribution mechanisms for wireless sensor networks," in *Proceedings of the 9th European Symposium On Research in Computer Security*, 2004.

[24] C. Castelluccia and C. Soriente, "ABBA: A balls and bins approach to secure aggregation in WSNs," in *6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, Berlin, Germany, Apr. 2008, pp. 185–191.

[25] C. Castelluccia, A. C.-F. Chan, E. Mykletun, and G. Tsudik, "Efficient and provably secure aggregation of encrypted data in wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, no. 3, pp. 1–36, 2009.

[26] C. Castelluccia, E. Mykletum, and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks," in *2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, San Diego, CA, USA, Jul. 2005, pp. 109–117.

[27] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: Application driver for wireless communications technology," in *ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean*, 2001.

[28] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *IEEE Symposium on Research in Security and Privacy*, 2003, pp. 197–213.

[29] ——, "Secure hierarchical in-network aggregation in sensor networks," in *Proceedings of 13rd ACM Conference on Computer and Communications Security*, Oct. 2006.

[30] J.-H. Chang and L. Tassiulas, “Maximum lifetime routing in wireless sensor networks,” *IEEE/ACM Transactions on Networking*, vol. 12, no. 4, pp. 609–619, Aug. 2004.

[31] A. Chaudhuri and R. Mukerjee, *Randomized Response: Theory and Techniques.* Marcel Dekker, Inc., September 1988.

[32] A. M. Chen and M. D. Scott, “Immunocamouflage: Prevention of transfusion-induced graft-versus-host disease via polymer grafting of donor cells,” *Journal of Biomedical Materials Research Part A*, vol. 67A, no. 2, pp. 626–636, 2003.

[33] J.-Y. Chen, G. Pandurangan, and D. Xu, “Robust computation of aggregates in wireless sensor networks: Distributed randomized algorithms and analysis,” in *Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, UCLA, Los Angeles, CA, USA, 2005.

[34] S. Coleri, S. Y. Cheung, and P. Varaiya, “Sensor networks for monitoring traffic,” in *Allerton Conference on Communication, Control and Computing*, 2004.

[35] W. Conner, T. F. Abdelzaher, and K. Nahrstedt, “Using data aggregation to prevent traffic analysis in wireless sensor networks,” in *International Conference on Distributed Computing in Sensor Systems*, 2006, pp. 202–217.

[36] J. Corburn, “Confronting the challenges in reconnecting urban planning and public health,” *American Journal of Public Health*, vol. 94, no. 4, pp. 541 – 549, Apr. 2004.

[37] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth, “On the Lambert w function,” *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, 1996.

[38] R. Cramer, I. Damgård, and S. Dziembowski, “On the complexity of verifiable secret sharing and multiparty computation,” in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, Portland, OR, USA, May 2000, pp. 325–334.

[39] CTIA, “CTIA, consumer info,” `http://www.ctia.org/consumer_info/index.cfm/AID/10323`, Accessed Aug 7, 2012.

[40] I. Damgard, M. Geisler, and M. Kroigard, “Homomorphic encryption and secure comparison,” *International Journal of Applied Cryptography*, vol. 1, no. 1, pp. 22–31, 2008.

[41] C. De Cannière, "Trivium: A stream cipher construction inspired by block cipher design principles," in *Proceedings of the 9th international conference on Information Security*, Samos Island, Greece, 2006, pp. 171–186.

[42] G. de Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communication*, Los Alamitos, CA, USA, 2008, pp. 580–585.

[43] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Compressed sensing," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[44] A. Deshpande, S. Nath, P. B. Gibbons, and S. Seshan, "Cache-and-query for wide area sensor databases," in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of data*, San Diego, CA, USA, 2003, pp. 503–514.

[45] R. Dingledine, N. Mathewson, and P. Syverson, "Deploying low-latency anonymity: Design challenges and social factors," in *Proceedings of the IEEE Symposium on Security & Privacy*, Oakland, CA, USA, Sep. 2007.

[46] J. Domingo-Ferrer, "A new privacy homomorphism and applications," *Information Processing Letters*, vol. 60, no. 5, pp. 277–282, 1996.

[47] N. I. Dopico, C. Gil-Soriano, I. Arrazola, and S. Zazo, "Analysis of IEEE 802.15.4 throughput in beaconless mode on MicaZ under TinyOS 2," in *Fall Vehicular Technology Conference*, 2010, pp. 1–5.

[48] L. Dóra and T. Holczer, "Hide-and-lie: Enhancing application-level privacy in opportunistic networks," in *Proceedings of the Second International Workshop on Mobile Opportunistic Networking*, Pisa, Italy, 2010, pp. 135–142.

[49] G. Dror, N. Koenigstein, Y. Koren, and M. Weimer, "The Yahoo! music dataset and KDD-Cup'11. KDD-Cup Workshop," 2011.

[50] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key predistribution scheme for wireless sensor networks," in *Proceedings of the 10th ACM Conference on Computer and Communications Security*, Washington, DC, USA, Oct. 2003, pp. 42–51.

[51] W. Du and Z. Zhan, "Using randomized response techniques for privacy-preserving data mining," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, D.C., 2003, pp. 505–510.

[52] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming, Part II*, Venice, Italy, Jul. 2006, pp. 1–12.

[53] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proceedings of the 3rd Theory of Cryptography Conference*, 2006, pp. 265–284.

[54] C. Dwork, M. Naor, T. Pitassi, G. Rothblum, and S. Yekhanin, "Pan-private streaming algorithms," in *Proceedings of the 1st Symposium on Innovations in Computer Science*, Beijing, China, Jan. 2010, pp. 66–80.

[55] L. Ertaul and V. Kedlaya, "Computing aggregation function minimum/maximum using homomorphic encryption schemes in wireless sensor networks (WSNs)," in *Proceedings of the 2007 International Conference on Wireless Networks*, 2007, pp. 186–192.

[56] L. Ertaul and J. H. Yang, "Implementation of Domingo Ferrer's a new privacy homomorphism (DF a new PH) in securing wireless sensor networks (WSNs)," in *Proceedings of the 2008 International Conference on Security & Management*, Las Vegas, NV, USA, Jul. 2008, pp. 498–504.

[57] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *In Proceedings of the 9th ACM Conference on Computer and Communications Security*, Nov. 2002, pp. 41–47.

[58] F. Esponda, "Negative representations of information," Ph.D. dissertation, University of New Mexico, 2005.

[59] ——, "Negative surveys," *ArXiv Mathematics e-Prints*, Aug. 2006.

[60] ——, "Everything that is not important: Negative databases," *IEEE Computational Intelligence Magazine*, May 2008.

[61] F. Esponda, E. S. Ackley, P. Helman, H. Jia, and S. Forrest, "Protecting data privacy through hard-to-reverse negative databases," *International Journal of Information Security*, vol. 6, no. 6, pp. 403–415, 2007.

[62] F. Esponda, S. Forrest, and P. Helman, "Enhancing privacy through negative representations of data," University of New Mexico, Tech. Rep., 2004.

[63] F. Esponda and V. M. Guerrero, "Surveys with negative questions for sensitive items," *Statistics & Probability Letters*, vol. 79, no. 15, pp. 2456–2461, Dec. 2009.

[64] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke, “Privacy preserving mining of association rules (invited journal version),” *Journal of Information Systems*, vol. 29, no. 4, pp. 343–364, Jun. 2004.

[65] T. Feng, C. Wang, W. Zhang, and L. Ruan, “Confidentiality protection schemes for data aggregation in sensor networks,” in *Proceedings of the 27th IEEE International Conference on Computer Communications*, Phoenix, AZ, USA, Apr. 2008.

[66] A. Fernandez-Montes, L. Gonzalez-Abril, J. Ortega, and F. Morente, “A study on saving energy in artificial lighting by making smart use of wireless sensor networks and actuators,” *IEEE Network*, vol. 23, no. 6, pp. 16–20, Nov.-Dec. 2009.

[67] J. A. Fox. and P. E. Tracy, *Randomized Response: A Method for Sensitive Surveys (Quantitative Applications in the Social Sciences).* Sage Publications, July 1986.

[68] FOXNEWS.com, “Homeland security looking into cell phones as anti-terror device,” 2007, `http://www.foxnews.com/story/0,2933,270033,00.html`.

[69] F. Furfaro, G. M. Mazzeo, and D. Saccà, “A probabilistic framework for building privacy-preserving synopses of multi-dimensional data,” in *Proceedings of the 20th International Conference on Scientific and Statistical Database Management*, Hong Kong, China, Jul. 2008, pp. 114–130.

[70] P. Ganesan, R. Venugopalan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu, “Analyzing and modeling encryption overhead for sensor network nodes,” in *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, San Diego, CA, USA, 2003, pp. 151–159.

[71] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher, “PoolView: Stream privacy for grassroots participatory sensing,” in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, Raleigh, NC, USA, 2008, pp. 281–294.

[72] F. Gil-Castineira, F. Gonzalez-Castano, R. Duro, and F. Lopez-Pena, “Urban pollution monitoring through opportunistic mobile sensor networks based on public transport,” in *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, Jul. 2008, pp. 70–74.

[73] J. Girao, D. Westhoff, and M. Schneider, “CDA: Concealed data aggregation for reverse multicast traffic in wireless sensor networks,” in *Proceedings of the*

40th IEEE International Conference on Communications, Seoul, Korea, May 2005.

[74] R. Goering, "Matlab edges closer to electronic design automation world," 2004, EE Times, `http://www.eetimes.com/electronics-news/4050334/Matlab-edges-closer-to-electronic-design-automation-world`, Accessed Mar. 29, 2012.

[75] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*. New York, NY: Cambridge University Press, 2004.

[76] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private Internet connections," *Communications of the ACM*, vol. 42, no. 2, Feb. 1999.

[77] Google, "Google powermeter," 2009, `http://www.google.org/powermeter/`.

[78] M. M. Groat, B. Edwards, J. Horey, W. He, and S. Forrest, "Enhancing privacy in participatory sensing applications with multidimensional data," in *Proceedings of the 10th Annual IEEE International Conference on Pervasive Computing and Communications*, Lugano, Switzerland, Mar. 2012, pp. 144–152.

[79] M. M. Groat, W. He, and S. Forrest, "KIPDA: k-Indistinguishable privacy-preserving data aggregation in wireless sensor networks," in *Proceedings of the 30th IEEE International Conference on Computer Communications*, Shanghai, China, Apr. 2011, pp. 2024–2032.

[80] M. Gruteser, G. Schelle, A. Jain, R. Han, and D. Grunwald, "Privacy-aware location sensor networks," in *Proceedings of the 9th conference on Hot Topics in Operating Systems*, Lihue, HI, USA, 2003, pp. 28–28.

[81] J. Halpern and V. Teague, "Rational secret sharing and multiparty computation: Extended abstract," in *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, Chicago, IL, USA, Jun. 2004, pp. 623–632.

[82] Harvard Sensor Networks Lab, "Codeblue: Wireless sensors for medical care," 2008, `http://fiji.eecs.harvard.edu/CodeBlue`.

[83] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: Privacy-preserving data aggregation in wireless sensor networks," in *26th Annual IEEE Conference on Computer Communications*, Anchorage, Alaska, 2007, pp. 2045–2053.

[84] J. Horey, S. Forrest, and M. Groat, "Reconstructing spatial distributions from anonymized locations," in *ICDE Workshop on Secure Data Management on Smartphones and Mobiles*, Washington D.C., April 2012.

[85] J. Horey, M. M. Groat, S. Forrest, and F. Esponda, "Anonymous data collection in sensor networks," in *4th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, Philadelphia, PA, USA, Aug. 2007, pp. 1–8.

[86] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*. Cambridge, UK: Cambridge University Press, 1991, ch. 4: Matrix equations and the Kronecker product, pp. 239–297.

[87] D. Horvitz, B. Shah, and W. Simmons, "The unrelated question randomized response model," in *Proceedings of the Social Statistics Section, American Statistical Association*, 1967.

[88] Q. Huang, H. J. Wang, and N. Borisov, "Privacy-preserving friends troubleshooting network," in *Proceedings of the 12th Annual Symposium on Network and Distributed Systems Security*, San Diego, CA, USA, Feb. 2005, pp. 245–257.

[89] Z. Huang and W. Du, "OptRR: Optimizing randomized response schemes for privacy-preserving data mining," in *Proceedings of the IEEE 24th International Conference on Data Engineering*, Cancun, Mexico, Apr. 2008, pp. 705–714.

[90] Z. Huang, W. Du, and B. Chen, "Deriving private information from randomized data," in *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, Baltimore, MD, USA, Jun. 2005, pp. 37–48.

[91] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of network density on data aggregation in wireless sensor networks," in *Proceedings of the 22nd International Conference on Distributed Computing Systems*, 2002.

[92] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, Atlanta, GA, USA, 2002.

[93] C. Jing, D. Shu, and D. Gu, "Design of streetlight monitoring and control system based on wireless sensor networks," in *2nd IEEE Conference on Industrial Electronics and Applications*, May 2007, pp. 57–62.

[94] R. Jurdak, P. Baldi, and C. V. Lopes, “Energy aware low power listening for sensor networks,” in *Second International Workshop on Networked Sensing Systems*, Jun. 2005.

[95] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar, “On the privacy preserving properties of random data perturbation techniques,” in *Proceedings of the 3rd IEEE International Conference on Data Mining*, Melbourne, FL, Nov. 2003, pp. 99–106.

[96] C. Karlof, N. Sastry, and D. Wagner, “TinySec: a link layer security architecture for wireless sensor networks,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, Baltimore, MD, USA, 2004, pp. 162–175.

[97] K. Kifayat, M. Merabti, Q. Shi, and D. Llewellyn-Jones, *Handbook of Information and Communication Security.* Springer Berlin Heidelberg, 2010, ch. Security in Wireless Sensor Networks, pp. 513–552.

[98] J. Ko, C. Lu, M. Srivastava, J. Stankovic, A. Terzis, and M. Welsh, “Wireless sensor networks for healthcare,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1947 –1960, Nov. 2010.

[99] B. Lai, S. Kim, and I. Verbauwhede, “Scalable session key construction protocol for wireless sensor networks,” in *IEEE Workshop on Large Scale RealTime and Embedded Systems*, Dec. 2002.

[100] X. Lai and J. L. Massey, “A proposal for a new block encryption standard,” in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, Aarhus, Denmark, 1991, pp. 389–404.

[101] S. H. Lee, S. Lee, H. Song, and H. S. Lee, “Wireless sensor network design for tactical military applications: Remote large-scale environments,” in *Proceedings of the 28th IEEE Conference on Military Communications*, Boston, Massachusetts, USA, 2009, pp. 911–917.

[102] M. LeMay, G. Gross, C. A. Gunter, and S. Garg, “Unified architecture for large-scale attested metering,” in *Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, Waikoloa, Big Island, HI, USA, Jan. 2007.

[103] P. Levis, N. Lee, M. Welsh, and D. Culler, “TOSSIM: Accurate and scalable simulation of entire TinyOS applications,” in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, Los Angeles, CA, USA, 2003, pp. 126–137.

[104] M. Li and Y. Liu, “Underground structure monitoring with wireless sensor networks,” in *6th International Symposium on Information Processing in Sensor Networks*, Cambridge, MA, USA, Apr. 2007.

[105] Y. Lindell and B. Pinkas, “Privacy Preserving Data Mining,” *Journal of Cryptology*, vol. 15, no. 3, pp. 177–206, 2002.

[106] A. Liu and P. Ning, “TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks,” in *Proceedings of the 7th international conference on Information processing in sensor networks*, Washington, DC, USA, 2008, pp. 245–256.

[107] D. Liu and P. Ning, “Establishing pairwise keys in distributed sensor networks,” in *Proceedings of 10th ACM Conference on Computer and Communications Security*, Oct. 2003, pp. 52–61.

[108] L. Lu, J. Han, L. Hu, Y. Liu, and L. M. Ni, “Dynamic key-updating: Privacy-preserving authentication for RFID systems,” in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications*, White Plains, NY, Mar. 2007, pp. 13–22.

[109] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “The design of an acquisitional query processor for sensor networks,” in *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, 2002, pp. 491–502.

[110] ——, “TAG: A tiny aggregation service for ad-hoc sensor networks,” *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 131–146, Dec. 2002.

[111] Mathworks, “MATLAB,” `http://www.mathworks.com/products/matlab/`, Accessed Aug. 7, 2012.

[112] J. Meyerowitz and R. Roy Choudhury, “Hiding stars with fireworks: Location privacy through camouflage,” in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, Beijing, China, Sep. 2009, pp. 345–356.

[113] A. Meyerson and R. Williams, “On the complexity of optimal k-anonymity,” in *Proceedings of the 23rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, Paris, France, Jun. 2004, pp. 223–228.

[114] F. P. Miller, A. F. Vandome, and J. McBrewster, *Advanced Encryption Standard.* Alpha Press, 2009.

[115] J. J. A. Moors, "Optimization of the unrelated question randomized response model," *Journal of the American Statistical Association*, vol. 66, no. 335, pp. 627–629, 1971.

[116] K. Motoi, M. Ogawa, H. Ueno, Y. Kuwae, A. Ikarashi, T. Yuji, Y. Higashi, S. Tanaka, T. Fujimoto, H. Asanoi, and K.-i. Yamakoshi, "A fully automated health-care monitoring at home without attachment of any biological sensors and its clinical evaluation," in *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Sep. 2009, pp. 4323–4326.

[117] National Institute of Standards and Technology (NIST), "SKIPJACK and KEA Algorithm Specifications Version 2.0," National Institute of Standards and Technology (NIST), Tech. Rep., May 1998.

[118] Network Working Group, "TEP 123: The collection tree protocol (CTP)," 2006, `www.tinyos.net/tinyos-2.x/doc/html/tep123.html` Accessed Aug. 15, 2012.

[119] M. Nishiyama, H. Sasaki, and K. Watanabe, "Wearable sensing clothes embedding a hetero-core optic fiber for recognizing arm segment posture and motion," in *5th IEEE Conference on Sensors*, Oct. 2006, pp. 1519–1522.

[120] V. G. Papanicolaou, G. E. Kokolakis, and S. Boneh, "Asymptotics for the random coupon collector problem," *Journal of Computional Applied Mathematics*, vol. 93, no. 2, pp. 95–105, 1998.

[121] G. Paul and S. Maitra, *RC4 Stream Cipher and Its Variants*, ser. Discrete Mathematics and Its Applications. Taylor & Francis, 2011.

[122] E. Perla, A. O. Catháin, R. S. Carbajo, M. Huggard, and C. Mc Goldrick, "PowerTOSSIM-z: Realistic energy modeling for wireless sensor network environments," in *Proceedings of the 3rd ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks*, Vancouver, British Columbia, Canada, 2008, pp. 35–42.

[123] A. Perrig, M. Luk, and C. Kuo, "Message-in-a-bottle: User-friendly and secure key deployment for sensor nodes," in *Proceedings of the ACM Conference on Embedded Networked Sensor System*, Oct. 2007.

[124] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, 2004.

[125] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," *Wireless Networks*, vol. 8, no. 5, pp. 521–534, Sep. 2002.

[126] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao, “CAP: A context-aware privacy protection system for location-based services.” in *Proceedings of the 29th International Conference on Distributed Computing Systems*, 2009, pp. 49 – 57.

[127] B. Pinkas, “Cryptographic techniques for privacy-preserving data mining,” *SIGKDD Explorations Newsletter*, vol. 4, no. 2, pp. 12–19, 2002.

[128] B. Przydatek, D. Song, and A. Perrig, “SIA: Secure information aggregation in sensor network,” *Proceedings of the ACM Conference on Embedded Networked Sensor Systems*, 2003.

[129] S. G. Quan and Y. Y. Kim, “Fast data aggregation algorithm for minimum delay in clustered ubiquitous sensor networks,” in *Proceedings of the International Conference on Hybrid Information Technology*, Los Alamitos, CA, USA, 2008, pp. 327–333.

[130] D. Quercia, I. Leontiadis, L. McNamara, C. Mascolo, and J. Crowcroft, “SpotME if you can: Randomized responses for location obfuscation on mobile phones,” in *Proceedings of the 31st International Conference on Distributed Computing Systems*, Minneapolis, MN, USA, 2011, pp. 363–372.

[131] R. L. Rivest, L. Adleman, and M. L. Dertouzos, “On data banks and privacy homomorphisms,” *Foundations of Secure Computation*, pp. 169–179, 1978.

[132] R. L. Rivest, “The RC5 encryption algorithm,” in *Proceedings of the Second International Workshop on Fast Software Encryption:*, Leuven, Belgium, Dec. 1994, pp. 86–96.

[133] P. Samarati and L. Sweeney, “Protecting privacy when disclosing information: k-Anonymity and its enforcement through generalization and suppression,” in *Proceedings of the IEEE Symposium on Research in Security and Privacy*, May 1998.

[134] C. Sharp, S. Schaffert, A. Woo, N. Sastry, C. Karlof, S. Sastry, and D. Culler, “Design and implementation of a sensor network system for vehicle tracking and autonomous interception,” in *Proceeding of the 2nd European Workshop on Wireless Sensor Networks*, Istanbul, Turkey, Jan.–Feb. 2005, pp. 93–107.

[135] E. Shi and A. Perrig, “Designing secure sensor networks,” *IEEE Wireless Communications*, vol. 11, no. 6, pp. 38–43, 2004.

[136] K. Singh and V. Muthukkumarasamy, “Information assurance protocols for body sensors using physiological data,” in *Biosensors*. InTech, 2010.

[137] I. Solis and K. Obraczka, "The impact of timing in data aggregation for sensor networks," in *2004 IEEE International Conference on Communications*, vol. 6, Paris, France, 2004, pp. 3640–3645.

[138] N. Subramanian, K. Yang, W. Zhang, and D. Qiao, "ElliPS: A privacy preserving scheme for sensor data storage and query," in *Proceedings of the 28th IEEE International Conference on Computer Communication*, Rio de Janeiro, Brazil, Apr. 2009, pp. 936–944.

[139] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, Oct. 2002.

[140] A. Tanenbaum, *Computer Networks*, 4th ed. Prentice Hall Professional Technical Reference, 2002.

[141] X. Tang and J. Xu, "Extending network lifetime for precision-constrained data aggregation in wireless sensor networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications*, 2006.

[142] B. L. Titzer and J. Palsberg, "Nonintrusive precision instrumentation of microcontroller software," in *Proceedings of the 2005 ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, Chicago, IL, USA, 2005, pp. 59–68.

[143] E. Trias, J. A. Navas, E. S. Ackley, S. Forrest, and M. V. Hermenegildo, "Negative ternary set-sharing," in *4th International Conference on Logic Programming*, Udine, Italy, 2008, pp. 301–316.

[144] J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, *Wireless sensor network security: A survey.* CRC Press, 2007, ch. 17.

[145] A. S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, Kauai, HI, USA, 2005, pp. 324–328.

[146] H. Wang, B. Sheng, and Q. Li, "Elliptic curve cryptography-based access control in sensor networks," *International Journal Security and Networks*, vol. 1, no. 3/4, pp. 127–137, 2006.

[147] S. L. Warner, "Randomized response: A survey technique for eliminating evasive answer bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, Mar. 1965.

[148] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "Monitoring volcanic eruptions with a wireless sensor network," in *Proceedings of the Second European Workshop on Wireless Sensor Networks*, Istanbul, Turkey, Jan.-Feb. 2005, pp. 108 – 120.

[149] D. Xiao, "Is privacy compatible with truthfulness?" Cryptology ePrint Archive, Tech. Rep. 2011/005, 2011, `http://eprint.iacr.org/`.

[150] H. Xie, L. Kulik, and E. Tanin, "Privacy-aware collection of aggregate spatial data," *Data & Knowledge Engineering*, vol. 70, no. 6, pp. 576–595, Jun. 2011.

[151] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: A secure hop-by-hop data aggregation protocol for sensor networks," in *The ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2006.

[152] A. C.-C. Yao, "Protocols for secure computations (extended abstract)," in *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, Chicago, IL, USA, Nov. 1982, pp. 160–164.

[153] Y. Yao, L. Ma, and J. Liu, "Privacy-preserving max/min aggregation in wireless sensor networks," *Advances in Information Sciences and Service Sciences*, vol. 4, no. 6, pp. 272–280, 2012.

[154] P. Zappi, E. Bales, J. H. Park, W. Griswold, and T. Šimunić Rosing, "The citisense air quality monitoring mobile sensor node," in *Proceedings of the 11th ACM/IEEE Conference on Information Processing in Sensor Networks*, Beijing, China, Apr. 2012.

[155] S. Zhang, J. Ford, and F. Makedon, "Deriving private information from randomly perturbed ratings," in *Siam Conference on Data Mining*, 2006.

[156] W. Zhang, C. Wang, and T. Feng, "GP2S: Generic privacy-preservation solutions for approximate aggregation of sensor data (concise contribution)," in *Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications*, Hong Kong, 2008, pp. 179–184.

[157] Y. Zhu and R. Sivakumar, "Challenges: Communication through silence in wireless sensor networks," in *Proceedings of the Eleventh Annual International Conference on Mobile Computing and Networking*, Cologne, Germany, 2005.

[158] D. Zurovac, A. O. Talisuna, and R. W. Snow, "Mobile phone text messaging: Tool for malaria control in Africa," *PLoS Medicine*, vol. 9, no. 2, Feb. 2012.