SANDIA REPORT

SAND2013-10675
Unlimited Release
Printed December 2013

Inherent Secure Communications Using Lattice Based Waveform Design

Matthew O. Pugh

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2013-10675
Unlimited Release
Printed December 2013

Inherent Secure Communications Using Lattice Based Waveform Design

Matthew O. Pugh (8229)

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-9999

Abstract

The wireless communications channel is innately insecure due to the broadcast nature of the electromagnetic medium. Many techniques have been developed and implemented in order to combat insecurities and ensure the privacy of transmitted messages. Traditional methods include encrypting the data via cryptographic methods, hiding the data in the noise floor as in wideband communications, or nulling the signal in the spatial direction of the adversary using array processing techniques.

This work analyzes the design of signaling constellations, i.e. modulation formats, to combat eavesdroppers from correctly decoding transmitted messages. It has been shown that in certain channel models the ability of an adversary to decode the transmitted messages can be degraded by a clever signaling constellation based on lattice theory. This work attempts to optimize certain lattice parameters in order to maximize the security of the data transmission. These techniques are of interest because they are orthogonal to, and can be used in conjunction with, traditional security techniques to create a more secure communication channel.

This work was funded by the Laboratory Directed Research and Development (LDRD) office at Sandia National Laboratories. Sandia is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Acknowledgment

This two-year project has been very challenging, yet exciting. While it has been mostly an individual effort, as many early career projects are, I would like to thank a few people. First I would like to thank my manager Paul Yoon for encouraging me to apply to the early career LDRD and supporting me throughout the process. Next I would like to thank Jacques Kvam, Derek Young and Ratish Punnoose for many interesting discussions. Lastly, I would like to thank Bill Ballard for his help and advice during the project.

This work was funded under LDRD Project Number 161862 and title "Inherent Secure Communications Using Lattice Based Waveform Design."

Contents

Nomenclature . . . 7
1 Introduction . . . 9
1.1 Fundamental Channel Model and Decoding Errors . . . 9
1.2 Lattices as Signaling Constellations . . . 11
1.3 Difficulties and Scope . . . 14
1.4 Outline . . . 15
2 Optimal Single Decision Regions . . . 17
3 Modulo Additive Noise Channel . . . 19
4 Integer Lattices . . . 25
4.1 Two Dimension Integer Lattice Analysis . . . 26
4.2 Higher Dimension Integer Lattices . . . 29
4.3 Realizability . . . 31
5 Newton's Method for Lattice Scaling . . . 33
5.1 Equivalence of Scaling Integration Region and Variance . . . 33
5.2 Scaling the One-Dimensional Lattice . . . 34
5.3 N-Dimensional Lattice Scaling . . . 36
5.4 Notes on Computing Newton's Method . . . 37
6 Random Lattices: Construction A . . . 39
6.1 Construction A Lattices . . . 39
6.2 Properties of Construction A Lattices . . . 41
6.3 Semidefinite Programming and Construction A Lattice Optimization . . . 46
7 Optimization of Lattices . . . 49
7.1 Positive Definite Quadratic Forms and Lattices . . . 49
7.2 Statistical Physics Approach . . . 53
7.3 Applications to Secure Communications . . . 53
8 Conclusion . . . 55
References . . . 57

Appendix

A Semidefinite Program to Optimize Construction A Lattices . . . 61

Figures

1 AWGN Eavesdropper Channel . . . 9
2 4-QAM Signaling Constellation . . . 10
3 Two Dimensional Gaussian PDF . . . 11
4 Examples of Two-Dimensional Lattices . . . 12
5 Coset Coding . . . 13
6 Probability of Correct Coding for Eve for QAM and Coset Decoding . . . 14
7 Convex Bodies and Gaussian Measure Initialized with $\alpha = 0.99$ . . . 17

8 Concentration of Measure Phenomenon 18
9 Modulo Additive Noise Channel 19
10 Decision Regions for Modulo Decoding and QAM Decoding 20
11 Eve's Performance Using Modulo Decoding and Coset Decoding 21
12 Eve's Performance Using Modulo Decoding and Coset Decoding 22
13 Two-Dimensional Signal Constellations 25
14 Translated Two Dimensional Integer Lattice 26
15 4-QAM Coset Coding 27
16 Edge Length β as a function of Probability Constraint α with $\sigma^2 = 1$ 28
17 Probability of Eve Successfully Decoding 29
18 Probability of Eve Successfully Decoding vs. α and σ^2_{Eve} 30
19 Edge Length β as a function of Probability Constraint α with $\sigma^2 = 1$ for several dimensions 31
20 Probability of Eve Successfully Decoding with $\sigma^2_{Bob} = 1$ 32
21 Gaussian Probability of Interval [-1,1] as a Function of β 35
22 Scaling Samples For Probability Constraint $\alpha = 0.90$ 38
23 Example of a Construction A Lattice for $n = 2$, $p = 7$ and $k = 1$ 40
24 Lattices with the same Performance but Different Generators 42
25 Number of Equivalence Classes for $n = 2$, $k = 1$ as a Function of prime p 43
26 Unique Construction A Lattices for $n = 2$, $k = 1$, $p = 7$ 43
27 Probability of Eve Correctly Decoding for Each Geometric Equivalence Class: $n = 2, k = 1, p = 7$ 44
28 Minimum Distance Distributions 44
29 Modulo-Gauss Measure 45
30 SDP Lattice Output for $n = 2, p = 5$ 48
31 Ellipse from the equation $x_1^2 + x_1x_2 + x_2^2$ 50

Nomenclature

AWGN additive white Gaussian noise

LDRD Laboratory Directed Research and Development

EC-LDRD Early Career Laboratory Directed Research and Development

QAM quadrature amplitude modulation

pdf probability density function

dB decibels

MIMO multiple-input multiple-output

PQF positive definite quadratic form

i.i.d. independent and identically distributed

SDP semidefinite programming

EDM Euclidean distance matrix

This page is intentionally left blank.

1 Introduction

The fundamental question addressed by this Early Career Laboratory Directed Research and Development (EC-LDRD) is how to design a signaling constellation to make it as difficult as possible for an adversary to intercept and successfully decode a transmitted message. It has been recently shown that using lattices to design the signaling constellation can help attain this goal. Before describing how lattices hinder eavesdroppers, the most fundamental channel model will be described as well as how decoding errors occur with traditional modulation methods.

1.1 Fundamental Channel Model and Decoding Errors

The Additive White Gaussian Noise Eavesdropper Channel

The channel model that will be considered throughout this LDRD is the additive white Gaussian noise (AWGN) channel with an eavesdropper. Figure 1 graphically illustrates this channel.

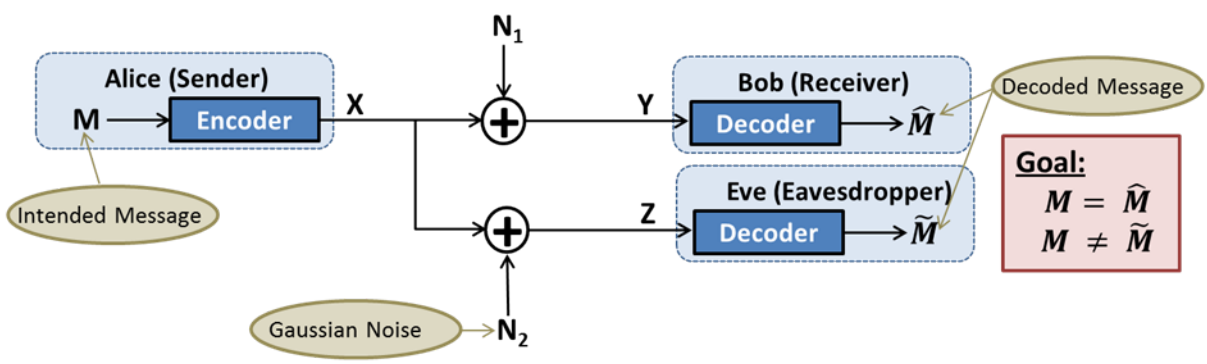


Figure 1. AWGN Eavesdropper Channel

Using the traditional nomenclature from the secure communication community, the transmitter is called Alice, the intended receiver is named Bob and the unwanted eavesdropper is named Eve. In this channel model, Alice sends a message $\mathbf{M}$ which is encoded as X and transmitted over the wireless channel. $\mathbf{X}$ is then corrupted by AWGN $\mathbf{N}_1$ as the it makes its way to Bob and is corrupted by AWGN $\mathbf{N}_2$ on its way to Eve. Here $\mathbf{N}_1$ and $\mathbf{N}_2$ are independent Gaussian random variables, i.e. $\mathbf{N}_1 \sim N\left(0, \sigma_{Bob}^2\right)$ and $\mathbf{N}_2 \sim N\left(0, \sigma_{Eve}^2\right)$ where $\mathbf{N}_1$ has variance σ_{Bob}^2 and $\mathbf{N}_2$ has variance σ_{Eve}^2. Bob thus receives a message $\mathbf{Y} = \mathbf{X} + \mathbf{N}_1$ which is decoded as $\hat{\mathbf{M}}$ and Eve receives a message $\mathbf{Z} = \mathbf{X} + \mathbf{N}_2$ which is decoded as $\tilde{\mathbf{M}}$. The goal is to have $\hat{\mathbf{M}} = \mathbf{M}$ and $\tilde{\mathbf{M}} \neq \mathbf{M}$. When the goal is accomplished, Bob has successfully decoded the message M while Eve decodes to an incorrect message. Because the channel is probabilistic due to the AWGN, the events $\hat{\mathbf{M}} = \mathbf{M}$ and $\tilde{\mathbf{M}} \neq \mathbf{M}$ have probabilities associated with them. Thus, the objective of this research takes on the following more concrete formulation:

Objective: Minimize the probability that Eve correctly decodes the transmitted message $\mathbf{M}$ while maintaining a probability α that Bob correctly decodes the transmitted message $\mathbf{M}$.

The objective is formally equivalent to: $\min \Pr\left[\tilde{\mathbf{M}} = \mathbf{M}\right]$ such that $\Pr\left[\hat{\mathbf{M}} = \mathbf{M}\right] = \alpha$. The constraint on the probability of correct decoding at Bob is required otherwise the optimization problem is trivial. This constraint is also essential as we want to be able to maintain a certain probability of correct decoding as this is the whole point of communication. If the message $\mathbf{M}$ is viewed as a symbol, then α can be selected at a reasonable level, e.g. $\alpha = 0.9$, with the knowledge that error correcting codes will be used to make the ultimate decoding error much less. Under the constraint that Bob's decoding probability is fixed, the objective is to minimize the probability that Eve correctly decodes the message $\mathbf{M}$.

Now that the objective has been given a mathematical formulation, some conditions on the noise random variables are implicitly enforced. Under the AWGN eavesdropper channel model, it is required that the noise variance $\sigma^2_{Bob} < \sigma^2_{Eve}$, i.e. the eavesdropper channel is noisier than the intended channel. If this is not the case, then Eve will decode with a higher success probability than Bob. This condition is not too serious however in that it seems very likely that the Eve's channel is noisier than Bob's because Bob's channel should be optimized.

Decoding Errors

With the condition that $\sigma^2_{Bob} < \sigma^2_{Eve}$, it will always be the case that $\Pr\left[\tilde{\mathbf{M}} = \mathbf{M}\right] \leq \Pr\left[\hat{\mathbf{M}} = \mathbf{M}\right]$ in the AWGN channel. The goal is to try to minimize the left hand side of this inequality. To gain insight into decoding behavior, consider the 4-QAM (quadrature amplitude modulation) case as illustrated in Figure 2. Suppose that the symbol 00 is transmitted but the point indicated by

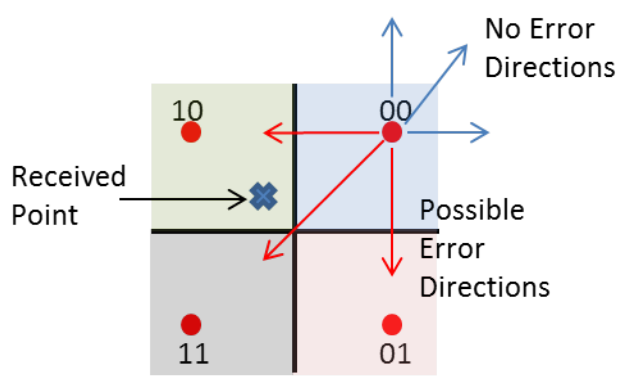


Figure 2. 4-QAM Signaling Constellation

the blue x is received. Under the AWGN channel, the decisions boundaries correspond with the coordinate axes (and are color coded in the image for convenience), so in this case the symbol 10 is decoded, which is an error. Still assuming that 00 was transmitted, the only way an error is made is if the noise makes the received symbol cross the decision boundary to the left or the decision

boundary below. Thus the noise vector must be in the directions of the red arrows in the figure. If the noise vector takes the received symbol in the directions of the blue arrows, no matter how far, no error is made since the decision region is unbounded. Under the AWGN model, all directions are equally likely and correspond to the circular equal probability contours of the Gaussian probability density function (pdf), as seen in Figure 3. As the noise variance of the Gaussian noise random variables is increased, the probability that the received symbol will cross a decision boundary is increased and corresponds to a flattening of the Gaussian pdf. Increasing the variance however does not change the fact that there are only certain directions that induce an error. The idea of using a lattice is that a (countably) infinite number of symbols are used so that every decision region is bounded and thus all directions can viably lead to an error. While trying to induce more errors may seem counterintuitive, this is important because in the eavesdropper channel, Eve has more noise variance, and thus it is easier for Eve to make a decoding error if there are more ways in which to make such an error. This reasoning will now be made more rigorous.

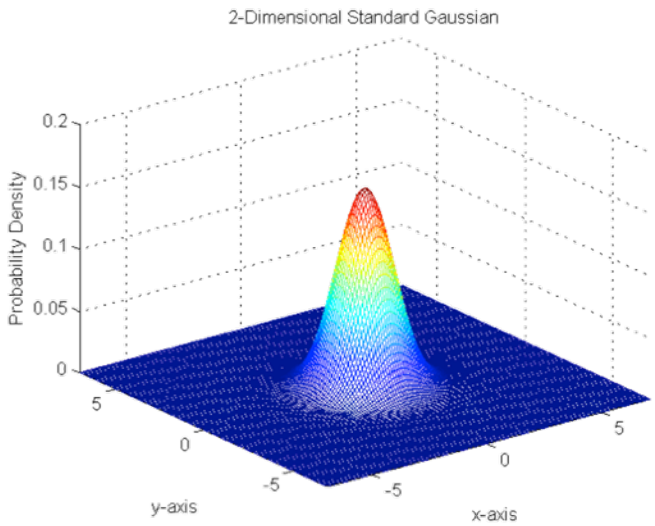
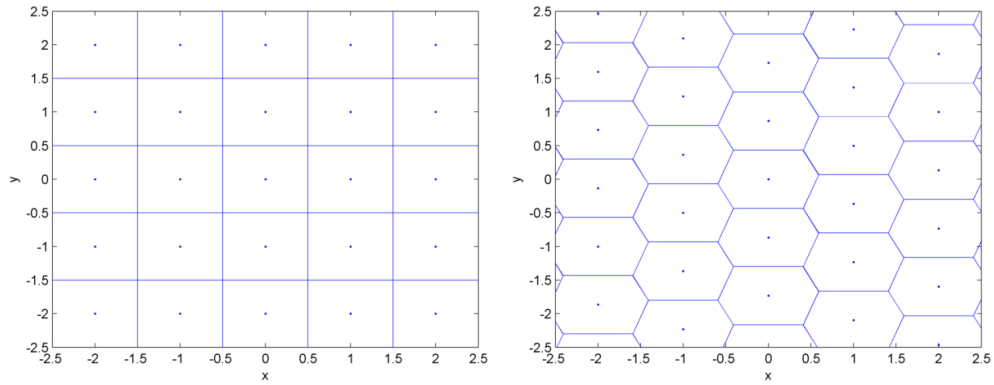


Figure 3. Two Dimensional Gaussian PDF

1.2 Lattices as Signaling Constellations

Before discussing the use of lattices in communications, a lattice needs to be defined. Formally, a lattice is a discrete additive group, i.e. the sum or difference of any two lattice points yields another lattice point, and in any given region of space, there are only finitely many lattice points. This description is vague, so two concrete examples are shown in Figure 4. Figure 4(a) is an example of an integer lattice where each coordinate of a lattice point $\mathbf{x} \in \mathbb{R}^2$ is an integer, e.g. $\mathbf{x} = \begin{bmatrix} 1 & 2 \end{bmatrix}^T$, where the superscript T denotes transpose. The lines in the lattices in Figure 4 partition the two-dimensional plane into Voronoi cells, or the points in the plane that are closest to each lattice point. A very important property of lattices is that the Voronoi cells are identical for

each lattice point. This is a very nice property because in the AWGN channel, the Voronoi cells are the optimal decision regions if a lattice point were transmitted, i.e. if the lattice were used as a signaling constellation. The lattice in Figure 4(b) is a hexagonal lattice whose Voronoi regions are hexagons.



(a) Integer Lattice

(b) Hexagonal Lattice

Figure 4. Examples of Two-Dimensional Lattices

One way to mathematically represent a lattices is via its generator matrix M. Given a full rank matrix $M \in \mathbb{R}^{n\times n}$ a lattice is defined by the set $\{\mathbf{y} = M\mathbf{x} | x \in \mathbb{Z}^n\}$ where n is the dimension and $\mathbf{x} \in \mathbb{Z}^n$ are all n-dimensional vectors with integer components. For the integer lattice in Figure 4(a) $M = \mathbf{I}$ the identity matrix and for the hexagonal lattice in Figure 4(b) $M = \begin{bmatrix} 1 & 0 \\ \frac{1}{2} & \frac{1}{2}\sqrt{3} \end{bmatrix}$. The generator matrix M is not unique for a given lattice, and in fact there are infinitely many generator matrices that produce the same lattice. Define the Gram matrix of a lattice as $A = M^T M$. The absolute value of the determinant of A, $|\det A|$ gives the volume of the Voronoi cells of the lattice. This value does not depend on which generator matrix M is used to describe the matrix.

Having briefly described some properties and given a couple examples of lattices, a connection will now be made to the communications. Looking at the 4-QAM constellation of Figure 2, it looks similar to the integer lattice from Figure 4(a). If the 4-QAM constellation is replicated and infinite number of times to generate the signaling constellation in Figure 5, then the signaling constellation is a translation of the integer lattice. We will now describe a technique known as coset coding ([18, 19, 20, 52]) using Figure 5 as an example. When the 4-QAM constellation was replicated, we also copy the bit-label given to each point. Thus, looking at the figure, all light-blue regions correspond to the bit label '00', all light green regions correspond to the bit label '11', etc. Thus, although there are an infinite number of code points, only 2 bits are being transmitted. Using coset coding, if '00' is to be transmitted, the transmitter is free to selected any of the lattice points corresponding to a light blue region. At first, this situation looks ridiculous as there is now a potential to select a code point with arbitrarily large energy. We will discuss how to by pass this problem in Section 3.

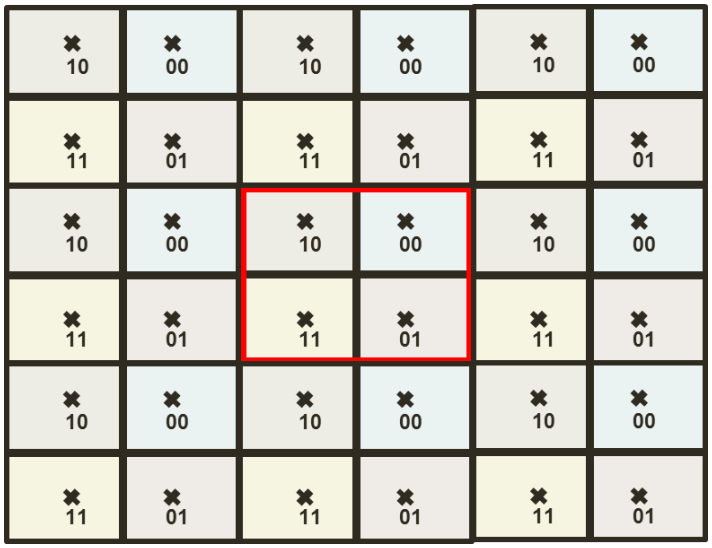


Figure 5. Coset Coding

For the time being, accepting this signaling constellation, what has been gained? As mentioned in the Section 1.1, only certain directions lead to decoding errors using 4-QAM in the AWGN channel. While the example was given for 4-QAM, any finite constellation has symbols which have unbounded decisions regions. When a lattice is used, each decision region, which is a Voronoi region in the AWGN channel, is bounded and therefore there are more directions or ways to make a decoding error. Assume that Bob's noise variance σ^2_{Bob} is very low and Eve's noise variance σ^2_{Eve} is large. For Bob, the probability of lying in the decision region of the transmitted code point is high since the noise variance is small. On the other hand, the probability that Eve's received point lies in the correct decision region (referring to Figure 5, this would be any of the regions sharing the same color as the transmitted code point) is small. This is also true in the QAM case, so how much is gained by the extra error directions? Figure 6 compares the probability that Eve correctly decodes the transmitted codeword as a function of Eve's variance σ^2_{Eve} for a fixed probability of Bob correctly decoding of $\alpha = 0.95$ and Bob's variance is fixed at $\sigma^2_{Bob} = 1$. There are a few things to notice in Figure 6. First is that when Eve has the same noise variance as Bob, predictably Bob and Eve have the same decoding performance. Secondly, and more importantly, notice that the probability of Eve correctly decoding decreases faster for the coset coding scheme (Figure 5) than for 4-QAM. This faster decay is due to the fact that there are more ways to make an error, as mentioned above. The details of how Figure 6 was generated will be covered in Chapter 4. Figure 6 motivates the user of lattices because for a given σ^2_{Eve}, $\Pr[\tilde{\mathbf{M}} = \mathbf{M}]$ should be minimized. Different lattice lead to different curves as found in Figure 6, so the question now becomes how to optimize the lattice to minimize the Eve's decoding probability. Lastly, Figure 6 indicates that there is a limit to how much the decoding probability can be decreased. In this example, the limit is 0.25, which makes sense: there are four possible codewords, and the worst one could do is to randomly guess which codeword was transmitted which yields a probability of correct decoding of $\frac{1}{4}$.

While this example is useful to gain insight into the fundamental aspects of the problem, there turns out to be nothing special about the two-dimensional case. Two-dimensional signaling constellations are common as the two axes correspond to the in-phase and quadrature components of a signal, but by concatenating these components, an arbitrarily large dimension can be attained. Thus, we can consider signaling constellations and lattices of arbitrary dimension.

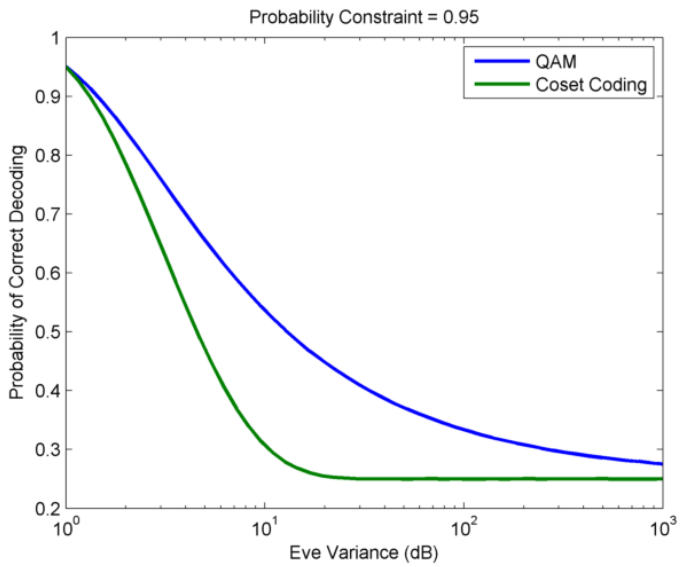


Figure 6. Probability of Correct Coding for Eve for QAM and Coset Decoding

1.3 Difficulties and Scope

The example comparing the decoding performance of an eavesdropper using 4-QAM and a two-dimension integer lattice with coset coding illustrates the fundamental problem addressed by this research. Under a given channel model, the probability of of Eve correctly decoding is to be minimized by optimizing the signal constellation over lattices. While it is easy to state the objective, it has turned out to be incredibly difficult to do in practice. The originating works in this area [6, 7, 8, 34] attacks this problem by approximating the Gaussian density with a second order Taylor series and using the Theta series of a lattice. This approach however is non-constructive as there is no way to convert an arbitrary Theta series into a lattice (i.e. one can compute a Theta series of a lattice, but given a Theta series one cannot invert it to produce a lattice). The original goal of this research was to have a more constructive approach so that theoretically a real system could be developed.

Under the AWGN channel model, the performance of a lattice in terms of providing secure communication will be shown to be related to the shape of the Voronoi regions. Specifically, simulations will show that generally the "more spherical" the Voronoi regions, the faster the rate

at which Eve's probability of correct decoding decreases. Making the Voronoi regions "more spherical" is related to maximizing the minimum distance between lattice points. This turns out to be a very difficult problem. In general finding the "best" lattice for different types of problems have puzzled mathematicians for thousands of years. For example, the optimal way to stack identical spherical oranges (a very practical problem done in every supermarket) in order to maximize the density of oranges was only solved less than 10 years ago and involved heavy computational effort ([22]). It is therefore not too surprising to find that explicit lattice optimization for secure communication is a challenging task. Outside of the AWGN channel, intuition into the relevant lattice parameters that need to be optimized is lacking. The original scope of the LDRD was to consider a variety of channel models, such as fading channels and Multiple-Input Multiple-Output (MIMO) channels. However, the extreme difficulty encountered in the simplest AWGN channel has precluded the analysis of these other channel models. This report will focus solely on the AWGN channel. Because of the difficulties in finding strictly optimum lattices, simulation, computational, and probabilistic arguments will be used to find good lattices for the use as signaling constellations.

1.4 Outline

A brief outline of this report will now be given. In Section 2, the shapes of a single decision region will be analyzed in terms of how fast the Gaussian probability escapes the region as a function of variance. These single decision regions may not correspond to a feasible lattice Voronoi regions, but insight will be gained into what shapes have good secure communication performance. As previously mentioned, the shape of a good decision region will be such that as the Gaussian noise variance increases, the probability inside the decision region decays as fast as possible. It will be shown that under the AWGN channel model, the more spherical the decision region, the faster the rate of decay. This leads to the intuition that the Voronoi regions of a good lattice should be as "spherical" as possible.

In Section 1.2 it is mentioned that naively using a lattice as a signaling constellation is not feasible because there are a countably infinite number of code points. Section 3 discusses a way to have a finite signaling constellation yet emulates the lattice. The central concept is to use a modulo additive noise channel, as used in [16]. The main idea is to use a random dither (common randomness) in association with the modulo operation to emulate an infinite number of lattice points.

Section 4 analyzes the only case that yielded analytical results: the integer lattice. Although this case is perhaps the most trivial, it is also the most realistic in terms of system realizability. The computations required to determine which Voronoi region a received code point lies in can be very complex. In the case of the integer lattice this computation is very straight forward since it can be done on a coordinate-by-coordinate basis. Thus this case is of interest from a practicality perspective.

One of the main difficulties in the computational analysis of lattice performance is determining the scaling of the lattice such that the Gaussian probability in a Voronoi region meets the probability

constraint $\Pr\left[\hat{\mathbf{M}} = \mathbf{M}\right] = \alpha$, i.e. that Bob can decode correctly with the desired probability. In Section 5 Newton's method will be used to quickly find the correct scaling of the lattice to meet the probability constraint. Prior to deriving this algorithm, a huge amount of computational time was required to determine the proper scaling. The use of this algorithm greatly increases the ability to simulate and test lattice performance.

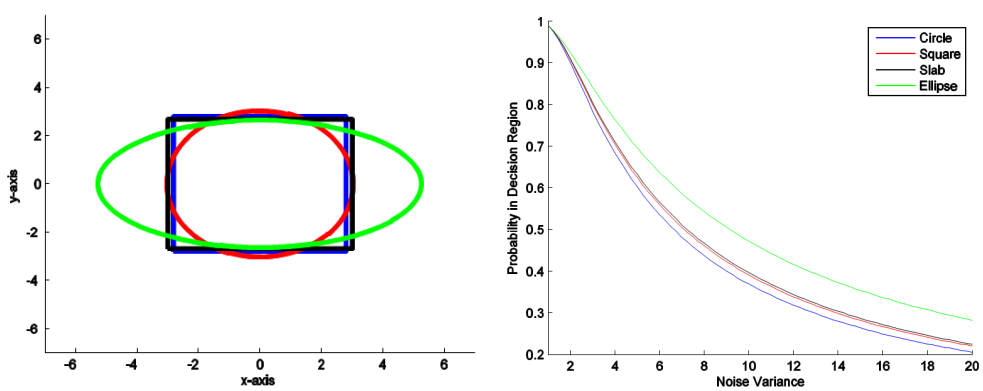
Because it is difficult to directly optimize lattices, a different approach is taken. Random lattices will be constructed from an ensemble of lattices which have previously been shown to have good characteristics for communications applications ([12, 16]). Simulations show the distribution of the maximum minimum distance of lattice lattice ensemble. Simulations will also show how these lattice perform in terms of hindering an eavesdropper. Some results on the use of semidefinite programming to optimize construction A lattices will also be presented. These results are limited but show the possibility of the use of the technique to optimize a class of lattices.

Section 7 discusses several possible techniques that could be used to find very good lattices for secure communications. Most of the work in this section has happened very recently and is thus preliminary, yet interesting. In fact, had there been more time, some of these techniques seem to have promise in the ability to find very good lattices. Very recently, the statistical mechanics community has been interested in finding the most dense lattices of a given dimension. Several algorithms have been proposed that, while not finding the densest lattice, find very dense lattices, in some cases the densest known lattice for a given dimension ([2, 23, 31, 35, 46, 47]). Due to time, these algorithms unfortunately have not been implemented to try to find good lattices for our problem, but it is believed that these algorithms could be successful, and thus a brief description of them is included. Lastly, Section 8 concludes the document by summarizing the findings and discussing possible future directions.

2 Optimal Single Decision Regions

Finding the best lattice for secure communications is a very challenging problem with no known optimal approach. To try to gain insight into the problem, this section investigates how the Gaussian probability inside a symmetric convex body (which all Voronoi decision regions are) changes as a function of the variance or equivalently as a function of scaling the size of the symmetric convex body (see Section 5.1 for details of this equivalence). To motivate this analysis, consider the case where $\sigma_{Bob}^2 = \sigma_{Eve}^2$. In this case, Eve performs just as well as Bob and is the worst case scenario (under the assumption $\sigma_{Eve}^2 \geq \sigma_{Bob}^2$) and given that α is near unity, nearly all of the probability mass lies in the decision region corresponding to the correct code point. For example, looking ahead at Figure 15, suppose '00' in the dark red region is to be transmitted and $\alpha = 0.99$, then nearly all of the Gaussian probability lies in the dark red decision region for '00' and all the coset equivalent regions in light red can be ignored. Now suppose that Eve's variance is increased slightly, i.e. $\tilde{\sigma}_{Eve}^2 = \sigma_{Eve}^2 + \varepsilon$ for some small $\varepsilon > 0$. What decision region shape leads to the biggest decrease in Eve's probability of correct decoding? It is this question that motivates the analysis of symmetric convex bodies and individual decision regions.

There is a substantial literature regarding the Gaussian probability inside convex bodies. See for example [9, 13, 21, 26, 27]. In this work, computer simulation are written to gain insight into this question. Figure 7(a) shows four convex bodies that are scaled such that the probability of a zero mean Gaussian random variable with covariance matrix $\mathbf{I}$ lying inside the bodies is equal to $\alpha = 0.99$. Figure 7(b) shows the probability of lying inside these four bodies as a function of the variance increasing. Notice that the probability decays the fastest for the circle and the slowest for



(a) Different Convex Bodies Containing the Same Gaussian Mass

(b) Gaussian Mass in Convex Body as a Function of Variance

Figure 7. Convex Bodies and Gaussian Measure Initialized with $\alpha = 0.99$

the ellipse. This figure seems to suggest that the more "spherical" the decision region, the faster the decay for Eve being able to correctly decode. Note that this figure is used to gain insight;

decision regions of a lattice are always polyhedral so a circular or elliptical decision region is not possible. It turns out that for a n-dimensional Gaussian a spherical convex body will have the fastest decay. This is because the equiprobability contours of the Gaussian density function in this case are spherical. Increasing the variance of the Gaussian density can be viewed as a probability flux of probability mass leaving the convex body. In this case the probability mass is moving perpendicular to the equiprobability contours at every point and thus maximizes the rate at which the probability is exiting the convex body.

Another point of view is that of the concentration of measure phenomenon ([43, 44]). For the case of i.i.d. n-dimensional Gaussian random variables, the concentration of measure phenomenon says that the the samples start to look more and more like they lie on the surface of a sphere as the dimension increases. Figure 8 shows i.i.d. Gaussian samples drawn from $N\left(\mathbf{0}, \frac{1}{n}\mathbf{I}\right)$ for low and high dimension (the high-dimensional image is a two-dimensional cross-section). Some concentration

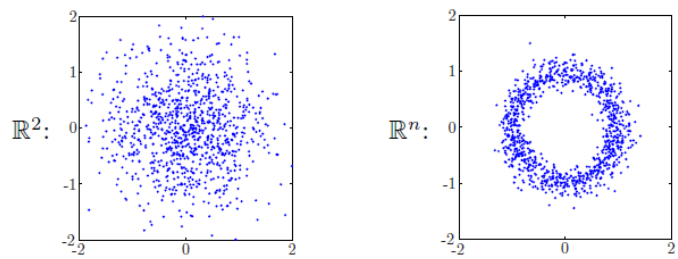


Figure 8. Concentration of Measure Phenomenon

equalities formalizing the intuition of the figure are now given (see [5] for details). Let $X \sim N(\mathbf{0}, \mathbf{I})$ be an n-dimensional Gaussian variable. Then

$$\Pr\left[\|\mathbf{X}\|^2 \geq \frac{n}{1-\varepsilon}\right] \leq e^{-\frac{\varepsilon^2 n}{4}} \tag{1}$$

$$\Pr\left[\|\mathbf{X}\|^2 \geq n(1-\varepsilon)\right] \leq e^{-\frac{\varepsilon^2 n}{4}} \tag{2}$$

$$\Pr\left[\|\mathbf{X}\|^2 \geq (1+\varepsilon)\mathbb{E}\left[\|\mathbf{X}\|^2\right]\right] \leq e^{-\frac{\varepsilon^2 n}{6}} \tag{3}$$

$$\Pr\left[\|\mathbf{X}\|^2 \leq (1-\varepsilon)\mathbb{E}\left[\|\mathbf{X}\|^2\right]\right] \leq e^{-\frac{\varepsilon^2 n}{4}}. \tag{4}$$

With these results in hand, it is conjectured that the best lattices for large α and $\sigma^2_{Eve} \approx \sigma^2_{Bob}$ have the most "spherical" decision regions. In Section 6, Figure 27 verifies this conjecture for a specific case. Based on this intuition, much of the work on this report is based on trying to find lattices with good parameters related to having "spherical" decision regions, such as having a large minimum distance between lattice points, which corresponds the being able to fit the largest possible sphere into the decision region.

3 Modulo Additive Noise Channel

In the introduction a brief description of lattices as signaling constellations and coset coding was discussed. It turns out that coset coding by itself is insufficient for the problem of trying to confuse any eavesdroppers. Before discussing the insufficiency, coset coding and the modulo additive noise channel will be discussed in more detail.

Consider again the example of a modified 4-QAM constellation. The right hand part of Figure 9 shows the four code points and the left hand side shows the copying and tiling of the code points and their decision regions to cover the plane. The fundamental region of a lattice point is the set of points in Euclidean space that are closer to the given lattice point than any other lattice point. For example, the light blue region represents the set of points in the plane that are closer to the code point '10' than any other code point. Notice that if we view the area enclosed in the bright red

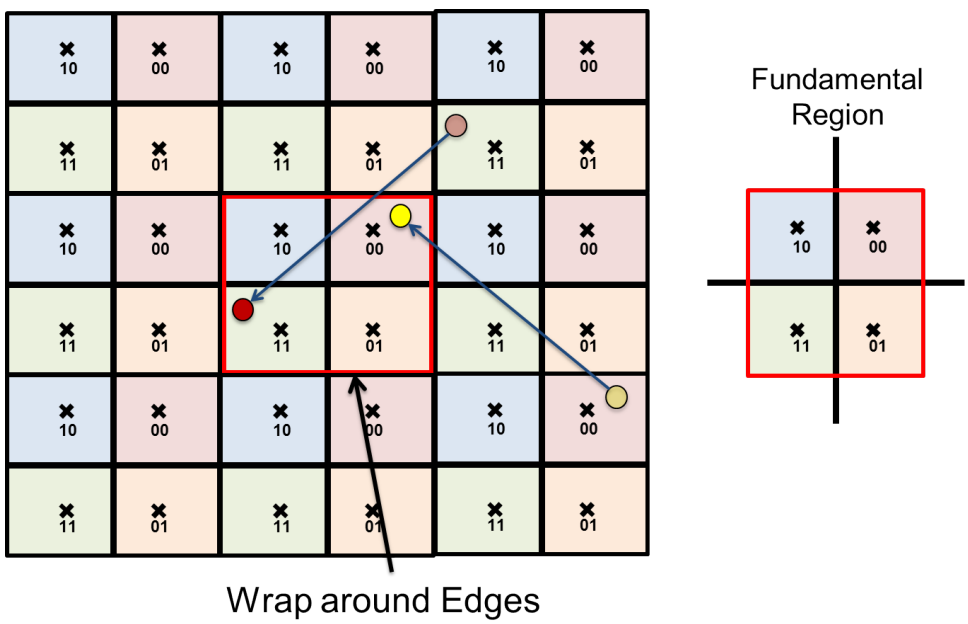
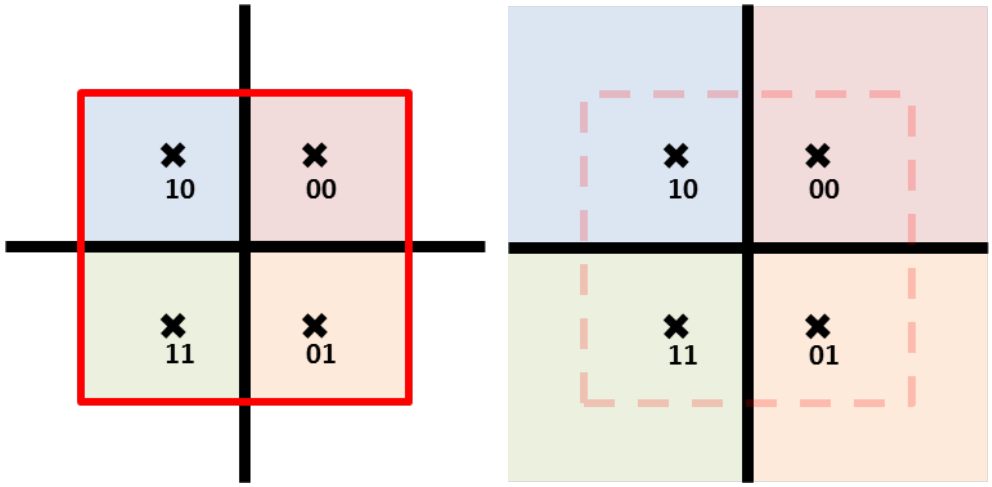


Figure 9. Modulo Additive Noise Channel

square as a fundamental region, then the tiling of the bright red squares generates another lattice. This "courser" lattice is a sub-lattice of the original lattice. The sub-lattice is courser in the sense that there is only a single square enclosed by the bright red square for every four smaller squares corresponding to the four code points.

The modulo additive noise channel intuitively works as follows. Select the base constellation of code points from a lattice. In Figure 9, the base constellation is 4-QAM. Replicate the base constellation and tile it to fill the Euclidean space. This generates a sub-lattice whose fundamental region is defined by the base constellation, which in the case of Figure 9 is the bright red square. Now suppose that the code point '00' is to be transmitted, call it **X** and suppose the noise vector

$\mathbf{N}$ translates $\mathbf{X}$ such that the received code point $\mathbf{Y} = \mathbf{X} + \mathbf{N}$ is the light red circle in the left hand side of the figure. Notice that the received code point lies in light green region corresponding to the decoding region for '11'. Instead of having to keep track of the entire two dimensional plane, an equivalent decoding procedure it to compute the received code point *modulo* the fundamental region of the sub-lattice, i.e. to wrap the received code point around the edges of the fundamental region of the sub-lattice until the code point lies in the fundamental region. In this example, wrapping the received code point (the light red circle) around the borders of the fundamental region (the bright red square) yields the code point designated by the bright red circle. Call the modulo received code point $\tilde{\mathbf{Y}}$. Notice that the bright red circle $\tilde{\mathbf{Y}}$ lies in the same position of the decision region for '11' as the original received code point, the light red circle $\mathbf{Y}$. It is in this sense that the modulo decoding procedure is equivalent to keeping track of all the decision regions in the plane, and thus using the modulo operation only the fundamental region of the sub-lattice (the bright red square) needs to be stored. As a second example, Figure 9 shows a received code point that is light yellow circle that is converted to the bright yellow circle and decoded as '00'. As is discussed in more detail in Section 4, the example given above is straight forward because the decision regions of the fundamental region of the sub-lattice (the bright red square) are parallel to the axes so that the modulo procedure can be done on each coordinate individually. For an arbitrary lattice and sub-lattice, this need not be the case and this can complicate the computation of the modulo operation.



(a) Modulo Decoding **(b)** QAM Decoding

Figure 10. Decision Regions for Modulo Decoding and QAM Decoding

If the entire lattice is used as a coding constellation, i.e. selecting any light blue point in the plane in Figure 9 to transmit '10', then Eve's performance degrades as a function of SNR faster than without the use of the lattice (e.g. just using QAM). This is the idea that originated the technique of coset coding [52]. However, as mentioned before, using the entire lattice requires

an infinite number of code points, which requires infinite power. It would be preferable to use the modulo technique described above as it has the same decoding regions but requires only the code points in the fundamental region of the sub-lattice (in the example above, the original 4-QAM points) and thus has finite power. Using the modulo constellation and modulo decoding as is does not work from a secure communications stand point. Just because Bob uses modulo decoding does not mean Eve does as well. Consider again the example given in Figure 9. The modulo decoding region, after performing the modulo operation on the received code point $\mathbf{Y}$ to yield $\tilde{\mathbf{Y}}$ is given in Figure 10(a). As is typically the case in analyzing eavesdropper performance, the worst case scenario is assumed. Suppose that Eve knows that Bob is using modulo decoding and has the constellation and decoding regions indicated in Figure 10(a). Rather than attempt to perform the same decoding, Eve can choose any decoding scheme she wishes, so assume that Eve uses a traditional QAM decoding scheme. The decision region for tradition 4-QAM is given in Figure 10(b). Note that the decision regions are unbounded. How does Eve's performance using QAM decoding compare to if she had used modulo decoding? Figure 11 gives the answer: Eve performs far better using just QAM decoding. In fact, for small values of σ^2_{Eve} (the case when $\sigma^2_{Eve} \approx \sigma^2_{Bob}$), Eve performs *better* than Bob. In this example Bob's probability constraint is $\alpha = 0.9$ and it can be seen that for small σ^2_{Eve}, Eve exceeds this probability. This is an extreme example but shows that *just* using modulo decoding is not enough to confuse an eavesdropper. What needs to be done in order to make Eve have the performance of the blue curve in Figure 11 rather than the red curve?

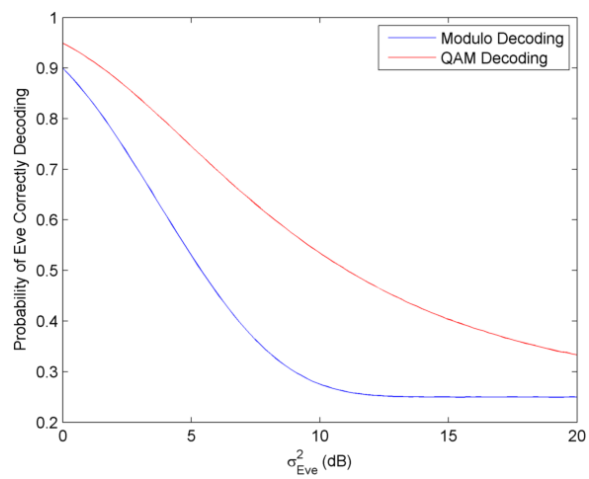


Figure 11. Eve's Performance Using Modulo Decoding and Coset Decoding

The key is to use a source of common randomness as is done in [16]. Let U be a random variable that is uniformly distributed over the volume of the fundamental region of the sub-lattice (e.g. U is randomly distributed over the bright red square in Figure 9). Assume that both the transmitter and receiver know U. Such a random variable is called common randomness and can be generated for example by having a random number generator at the receive and transmitter be initialized with the same seed. It is also assumed that Eve knows the common random variable U in order to analyze the worst-case scenario. If Eve does not know U, then Eve's probability

of correctly decoding can only be worse. Define the new transmitted code point with common randomness as $\mathbf{X}' = [\mathbf{X}+\mathbf{U}] \bmod \Lambda_{sub}$ where mod Λ_{sub} means the modulo operation with respect to the fundamental region of the sub-lattice. At the receiver, the received code point is

$$\mathbf{Y} = \mathbf{X}' + \mathbf{N} = [\mathbf{X}+\mathbf{U}] \bmod \Lambda_{sub} + \mathbf{N} \tag{5}$$

and the decoded code point is given by subtracting the common randomness and performing the modulo operation

$$\begin{aligned}\tilde{\mathbf{Y}}' &= [\mathbf{Y}-\mathbf{U}] \bmod \Lambda_{sub} \\ &= [[\mathbf{X}+\mathbf{U}] \bmod \Lambda_{sub} + \mathbf{N} - \mathbf{U}] \bmod \Lambda_{sub} \\ &= [\mathbf{X}+\mathbf{U}+\mathbf{N}-\mathbf{U}] \bmod \Lambda_{sub} \\ &= [\mathbf{X}+\mathbf{N}] \bmod \Lambda_{sub}\end{aligned} \tag{6}$$

where the identity from the second line to the third line is from the distributive property of the modulo operation. As defined in [16], the channel described by Equation 6 is called a *modulo-Λ additive noise channel*. Consider the same example as was discussed for Figures 10 and 11. Figure 12 compares four cases for Eve's decoding algorithm. The first two cases are the same as in Figure 11 and no common randomness is used. The two new cases consider a user of common randomness and compares Eve's decoding probability when she subtracts the assumed known common randomness (worst-case scenario) and performs the modulo operations (the same decoding procedure as Bob) and when Eve uses the standard QAM decision region on the received code point. Because of the equivalence of the modulo additive decoding with common randomness

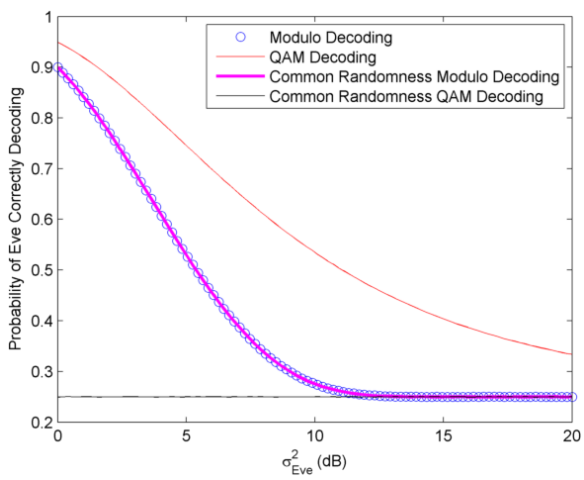


Figure 12. Eve's Performance Using Modulo Decoding and Coset Decoding

method in Equation 6 and the coset coding method described in Figure 9, the blue curve and magenta curve in Figure 12 are the same. If Eve uses only the QAM decision boundaries and ignores the common randomness (decoding procedure of the red curve), then Eve's probability of

successfully decoding the transmitted code word is greatly inhibited, and is in fact equal to random guessing.

The key take away from these examples is that the performance of coset coding with an infinite lattice, and thus infinite transmit power, can be reproduced using the equivalent modulo-Λ additive noise channel and common randomness which has a finite transmit power.

A Note on Analyzing Lattices

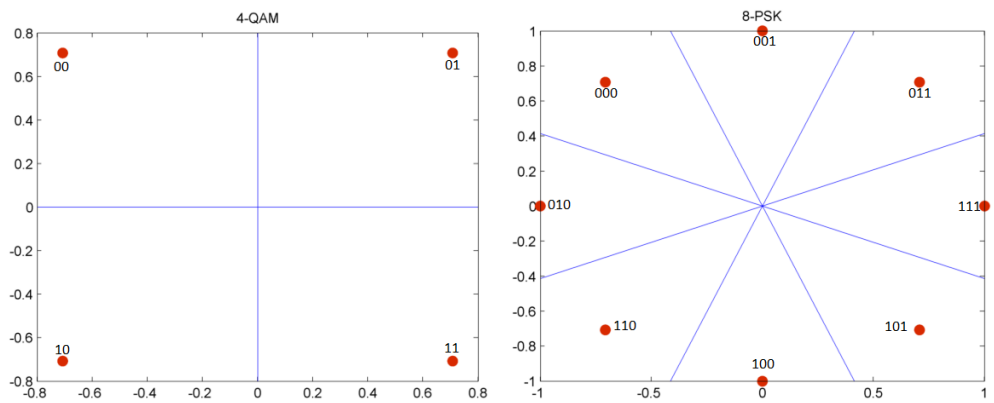
It has been shown that using coset coding without common randomness is suboptimal because it may be possible for Eve to use a different decoding algorithm and achieve a higher probability of correct decoding. To remedy this, common randomness is used to convert the channel with coset coding into the modulo-Λ additive noise channel. The analysis in the remainder of this report however will not use common randomness. System analysis will tend to be easier by not considering the common randomness, and it is known that equivalent results can be achieved by a conversion to the modulo-Λ additive noise channel.

This page is intentionally left blank.

4 Integer Lattices

This section analyzes the performance of integer lattices in terms of hindering the ability of an eavesdropper to correctly decode a transmitted message. The integer lattice is the easiest lattice to analyze due to the all the decision boundaries being parallel to a coordinate axis. Let $M \in \mathbb{R}^{n\times n}$ be the generator matrix for an n-dimensional lattice Λ, i.e. $\Lambda = M\mathbf{x}$ for all $\mathbf{x} \in \mathbb{Z}^n$. The generator matrix for an integer lattice is a scaled identity matrix, i.e. $M = \beta\mathbf{I}$ for some $\beta > 0$. All of the decision regions for the integer lattice are congruent hypercubes as seen in Figure 4(a). This is important, as will be seen latter, because computing the Gaussian probability of half-spaces is the only straight forward case, and hypercubes are the intersection of $2n$ half-spaces.

Typical two dimensional signaling constellations do not have a code point at the origin. For example, see the 4-QAM and 8-PSK constellations in Figure 13. All lattices contain the origin,



(a) 4-QAM **(b)** 8-PSK

Figure 13. Two-Dimensional Signal Constellations

so in order to make the integer lattice more like the 4-QAM constellation, the lattice must be translated. Comparing the integer lattice of Figure 4(a) with the 4-QAM constellation of Figure 13(a), it is clear that the translation vector is $\frac{\beta}{2}\mathbf{1}$, where β is the scaling constant and $\mathbf{1}$ is a vector of all ones. After translation, the lattice can be partition via coset coding to make, for example, 4-QAM as in Figure 5, or even higher order QAM constellations such as 16-QAM, 64-QAM, etc.

As mentioned earlier, the reason that integer lattices allow for amiable analysis is due to the error function and complementary error function which will now be described. Suppose $\mathbf{X}$ is a one dimensional Gaussian random variable with zero mean and unit variance. Consider events of the form $\mathbf{X} < c$ and $\mathbf{X} > c$, where c is a real number. What are the probabilities of these events? The one-dimensional Gaussian pdf is given by

$$f_{\mathbf{X}}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{7}$$

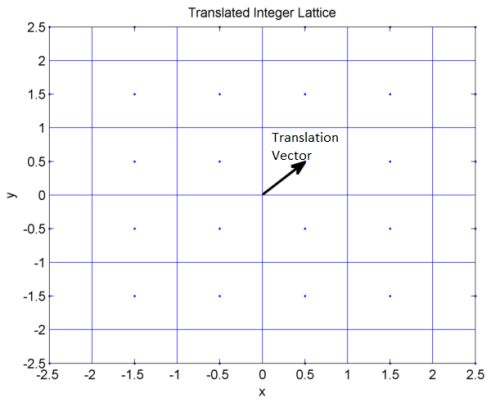


Figure 14. Translated Two Dimensional Integer Lattice

where μ is the mean and σ^2 is the variance. Thus, the probability of the event $\mathbf{X} < c$ is computed by

$$\Pr[\mathbf{X} < c] = \int_{-\infty}^{c} f_{\mathbf{X}}(x)dx = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{c} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx \tag{8}$$

and likewise $\mathbf{X} > c$ can be computed as

$$\Pr[\mathbf{X} > c] = \int_{c}^{\infty} f_{\mathbf{X}}(x)dx = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{c}^{\infty} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx. \tag{9}$$

These integrals have no closed-form solutions, but are well approximated and can be computed very accurately with the use of the error function and complementary error function. The error function is defined as

$$\mathrm{erf}(x) = \frac{2}{\pi} \int_{0}^{x} \exp(-t^2)dt \tag{10}$$

and the complementary error function is given by

$$\mathrm{erfc}(x) = 1 - \mathrm{erf}(x) = \frac{2}{\pi} \int_{x}^{\infty} \exp(-t^2)dt. \tag{11}$$

In the next section, these equations will be used to perform analysis of a two-dimension integer lattice.

4.1 Two Dimension Integer Lattice Analysis

The first task to be addressed is how to compute the proper scaling of the integer lattice β such that the constraint that the probability Bob correctly decodes the transmitted message is α is met. It is assumed that α is fairly large, i.e. fairly close to 1. In this case, the probability of a transmitted code point only lying in the originally transmitted decision region, and not all modulo-equivalent decision regions, needs to be computed. Figure 15 shows this graphically. Assuming the point 00

is transmitted, the probability of of the received code point lying in the dark red will be computed while all modulo equivalent regions in light red are ignored. The accuracy of the approximation by ignoring the light red regions increases as α approaches 1.

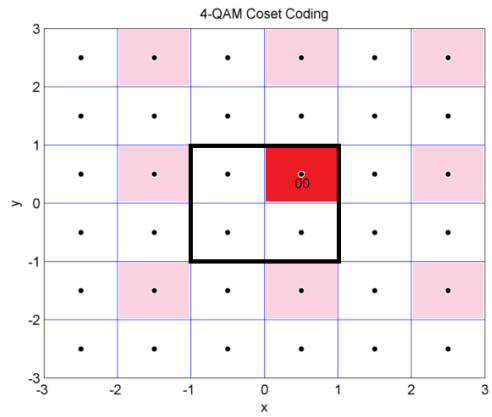


Figure 15. 4-QAM Coset Coding

In Figure 15, the transmitted code point is $\begin{bmatrix}\frac{1}{2} & \frac{1}{2}\end{bmatrix}^T$ and the decision region is the unit square. Let us assume the noise random variable $\mathbf{N} \sim N\left(0, \sigma^2\mathbf{I}\right)$. What β should the lattice be scaled by such that the probability of being in the dark red square (now having sides of length β) equals α? The key is to use the error function defined in Equation 10. For this analysis, without loss of generality, assume that the transmitted code point is the origin and compute the probability that the received code point lies in the box $\left[-\frac{\beta}{2}, \frac{\beta}{2}\right]^2$, i.e.

$$\begin{aligned}\Pr\left[-\frac{\beta}{2} \leq \mathbf{N}(1) \leq \frac{\beta}{2}\right] \wedge \Pr\left[-\frac{\beta}{2} \leq \mathbf{N}(2) \leq \frac{\beta}{2}\right] &= \Pr\left[-\frac{\beta}{2} \leq \mathbf{N}(1) \leq \frac{\beta}{2}\right]^2 \\ &= \left(2\Pr\left[0 \leq \mathbf{N}(1) \leq \frac{\beta}{2}\right]\right)^2 \end{aligned} \tag{12}$$

where $\mathbf{N}(1)$ is the first component of the noise vector and $\mathbf{N}(2)$ is the second component. The probability of being in the red square is equal to Equation 12 because each component of the noise is independent with equal variance, thus as long as both the first and second component lie in the interval, it lies in the square. The last equality is true because of the symmetry of the Gaussian density. The probability of being in the red square can be evaluated by use of the error function and a change of variables

$$\Pr\left[\mathbf{N} \in \text{Red Square}\right] = \left[\operatorname{erf}\left(\frac{\beta/2}{\sqrt{2\sigma^2}}\right)\right]^2 \tag{13}$$

The inverse error function is also well known, so the scaling factor and edge length can be solved by inverting the above Equation 13 and the approximation that only the red region matters:

$$\Pr\left[\mathbf{N} \in \text{Red Square}\right] = \alpha \Rightarrow \beta = 2\sqrt{2\sigma^2}\operatorname{erfinv}\left(\sqrt{\alpha}\right) \tag{14}$$

Figure 16 illustrates this function. Notice that as the probability constraint α approaches 1, the edge length approaches infinity. This makes sense because as α approaches 1, the red square needs to start covering most of the two dimensional plane in order to capture all of the probability.

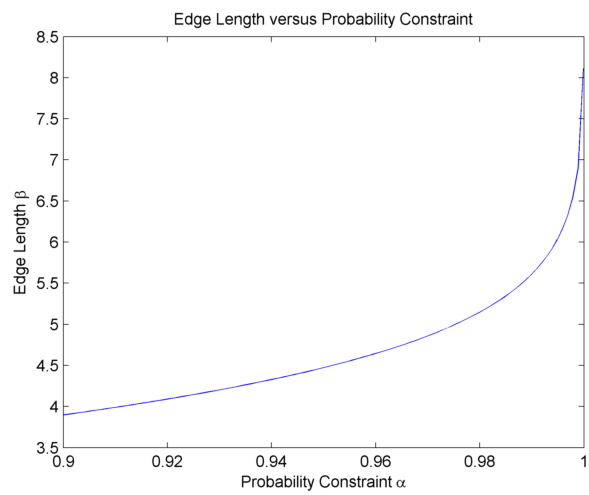


Figure 16. Edge Length β as a function of Probability Constraint α with $\sigma^2 = 1$

The next question to be addressed is how does Eve perform as a function of σ^2_{Eve} given the scaled integer lattice given by the solution of Equation 14? As an example, given the scaling β, the question is what is the probability that Eve's received codeword lies in *any* of the red regions (dark or light) in Figure 15? In theory, very good approximations can be achieved by using the error function and complementary error function, but accounting for all of the possible combination of outcomes quickly becomes tedious. Rather than take this approach, simulations will be performed on the modulo channel. For the time being assume that 2-bits are to be transmitted per symbol leading to the 4-QAM coset model as in Figure 15. Also assume that the transmitted codeword is $\begin{bmatrix} \frac{\beta}{2} & \frac{\beta}{2} \end{bmatrix}^T$. In this case, because all the decision boundaries are parallel to the coordinate axes, the modulo operation can be take component-by-component. Let $\mathbf{Z}$ be Eve's received codeword. The modulo equivalent channel takes each component of $\mathbf{Z}$ modulo 2β and call this result $\tilde{\mathbf{Z}}$, i.e. $\tilde{\mathbf{Z}} = \mathbf{Z} \mod 2\beta$. Eve decodes $\mathbf{X}$ correctly if $\tilde{\mathbf{Z}}$ lies in the square $[0, \beta]^2$. It is fairly easy to generate several million instances of $\mathbf{Z}$, covert it to $\tilde{\mathbf{Z}}$ and test if it lies in the square. The fraction of points lying in the square is an estimate of the probability of Eve successfully decoding. With several million samples, this estimate is accurate to a couple of decimal places, and the more samples that are used, the more accurate the estimate. Figure 17 shows the probability of correct reception as a function of Eve's variance (or equivalently SNR with respect to Bob's variance). In the figure, Bob's probability constraint was set to $\alpha = 0.9$ with $\sigma^2_{Bob} = 1$. When Eve's variance $\sigma^2_{Eve} = \sigma^2_{Bob}$, corresponding to the left hand side of the plot, Eve performs as well as Bob. In the blue curve, as Eve's variance increases, the probability of error decreases until it finally plateaus at a value of 0.25 corresponding to random guessing. If the coset signaling constellation has 16 points (4

bits), i.e. a four-by-four grid repeated instead of a 2x2 grid as in 16-QAM, the probability of correct detection decays to a value of $\frac{1}{16}$ for random guessing as seen in the red curve. Figure 17

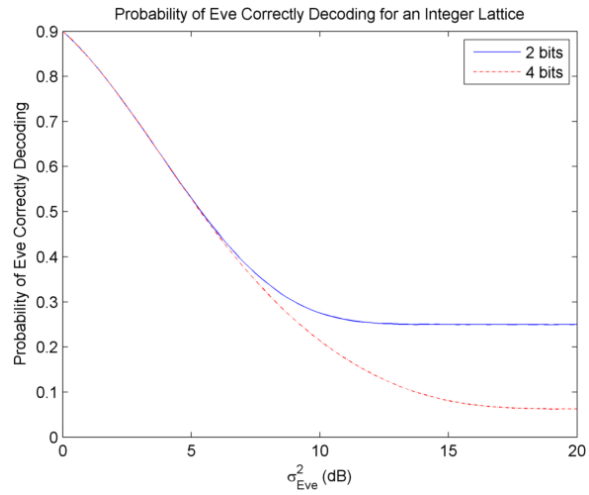


Figure 17. Probability of Eve Successfully Decoding

seems to suggest that by increasing the data rate, we hinder Eve's ability to correctly detect the transmitted signal. This is true, but as the number of bits per symbol is increased, the transmit energy is increased because of the constraint that Bob's probability of correct decoding equal α. In a traditional communications problem, it is usually the transmit power that is constrained and probability of decoding is maximized. In the system under consideration, increasing the bit rate eventually leads to an unrealizable transmit power.

Figure 18 shows the probability of Eve correctly decoding as a function of the probability constraint α and Eve's variance σ^2_{Eve}. The rate of decay as a function of σ^2_{Eve} is pretty similar in the range $[0.9, 0.995]$, but the smaller α is, the faster the decay. This is true up to a point, however, as α decreases the approximation by only the central decision region becomes less accurate and thus the rate of decay may not be accurate. In fact, it does not appear that the probability of Eve successfully decoding is monotonic in σ^2_{Eve} for certain α. However for large enough α, the probability of Eve successfully decoding is a monotonically decreasing function of σ^2_{Eve}.

4.2 Higher Dimension Integer Lattices

The computations in higher dimensions follows very closely with those in the two dimension case of the previous section. Under the same assumptions as the previous section, the probability of the received code point lying in the n-dimensional hypercube $\left[-\frac{\beta}{2}, \frac{\beta}{2}\right]^n$ given the origin was transmitted is

$$\Pr\left[-\frac{\beta}{2} \leq \mathbf{N}(1) \leq \frac{\beta}{2}\right]^n = \left(2\Pr\left[0 \leq \mathbf{N}(1) \leq \frac{\beta}{2}\right]\right)^n. \tag{15}$$

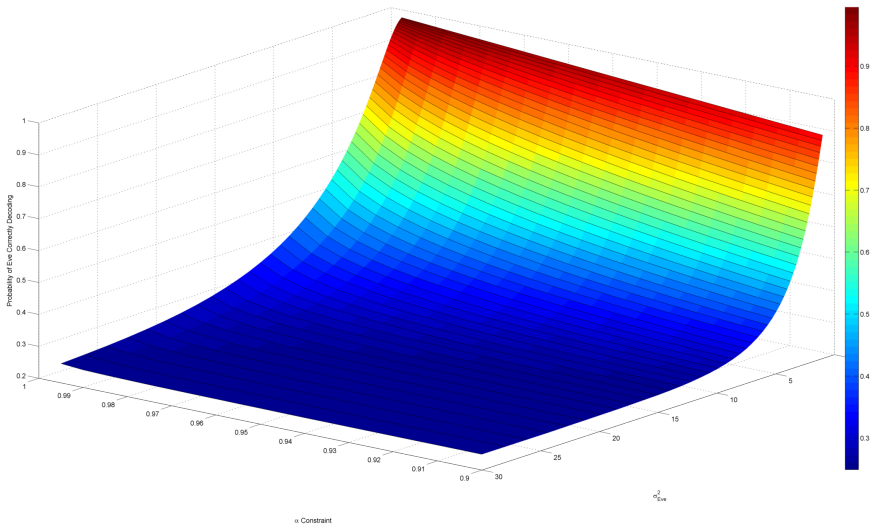


Figure 18. Probability of Eve Successfully Decoding vs. α and σ^2_{Eve}

This equation formally identical to Equation 12 with the exponent of 2 (corresponding to two dimensions) replaced by n. Similarly the higher dimension analogies of Equations 13 and 14 are:

$$\Pr\left[\mathbf{N} \in \text{Red Square}\right] = \left[\text{erf}\left(\frac{\beta/2}{\sqrt{2\sigma^2}}\right)\right]^n \tag{16}$$

and

$$\Pr\left[\mathbf{N} \in \text{Red Square}\right] = \alpha \Rightarrow \beta = 2\sqrt{2\sigma^2}\text{erfinv}\left(\alpha^{\frac{1}{n}}\right). \tag{17}$$

Figure 19 is analogous to Figure 18 except multiple dimensions are including. Notice that as the dimension increases, the edge length increasing to maintain a fixed α. This is due to the term $\alpha^{\frac{1}{n}}$ in the argument of the inverse error function. On the support of non-negative real numbers, the inverse error function is a monotonically increasing function of its argument. The probability constraint α lies in the interval $[0,1]$, so raising $\alpha^{\frac{1}{n}}$ is an increasing function of dimension n. Therefore, as the dimension increases the edge length increases.

An extension of the 4-QAM coset coding has two code points, $\pm\frac{\beta}{2}$, in each dimension for a total of 2^n code points. In theory the coset coding can involve any combination of fundamental regions such that the region covered by them is a rectangle. The problem with this general configuration is that the number of code points in the sub-lattice may not be a power of 2, and hence some bits are wasted. The beauty of generalized QAM configurations (4-QAM,16-QAM,etc.) is that the number of code points is always a power of 2 and thus no bits are wasted. Because of the independence of the noise, as was the situation in the the two dimensional case, each component of the n-dimensional Eve receive vector $\mathbf{Z}$ can be handled separately. Thus, the simulation technique used to generate Figure 17 can easily be generalized to n-dimensions. While in theory these techniques work for an arbitrary dimension, in practice the accuracy of our probability estimate decreases

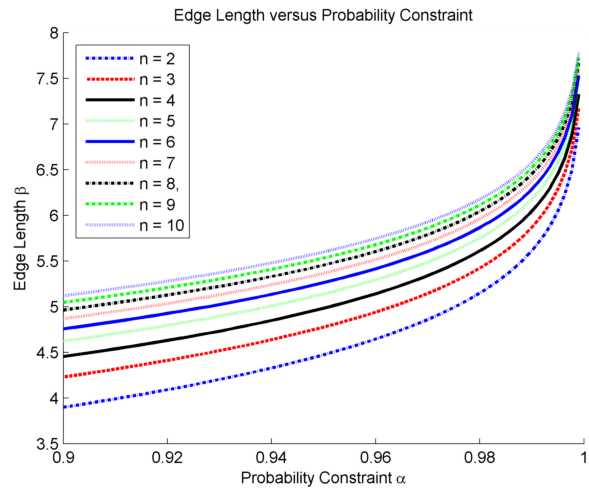


Figure 19. Edge Length β as a function of Probability Constraint α with $\sigma^2 = 1$ for several dimensions

because we need to generate more random variables and computer memory issues become a problem. Figure 20 is the n-dimensional analogy of Figure 17. As before, notice that when $\sigma^2_{Eve} = \sigma^2_{Bob}$, corresponding to the left hand side of the plot, Eve is able to decode correctly with the same probability as Bob. As Eve's variance σ^2_{Eve} is increased, the probability of correct decoding saturates. The saturation point is equal to the random guessing probability. As was previously mentioned, there are 2^n code points in the modified 4-QAM scheme, so the saturation probability is 2^{-n}. Figure 20 suggests that increasing the dimension hinders the ability of Eve to correctly decode. This is correct, but as discussed during the analysis of 4-QAM coset coding versus 16-QAM coset coding, the additional code points which help confuse Eve come at the cost of transmit power. This increase in transmit power can also be seen from Figure 19 where the edge length must increase to maintain a specified α, and the longer the edge width, the more power is required. To see this connection explicitly, in the n-dimensional 4-QAM extension, each code point is is of the form $\left[\pm\frac{\beta}{2},\ldots,\pm\frac{\beta}{2}\right]$. Thus, each code point has the same power and is equal to

$$\text{Code Point Power} = \sum_{i=1}^{n}\left(\pm\frac{\beta}{2}\right)^2 = n\frac{\beta^2}{4}. \tag{18}$$

Because each code point has the same power, the average transmit power equals the individual code point power. From this equation, it is easy to see that the transmit power increases linearly with dimension.

4.3 Realizability

In general, finding the nearest lattice point of an arbitrary point in Euclidean space is a challenging problem that has generated a large amount of research. It is known as the closest vector problem

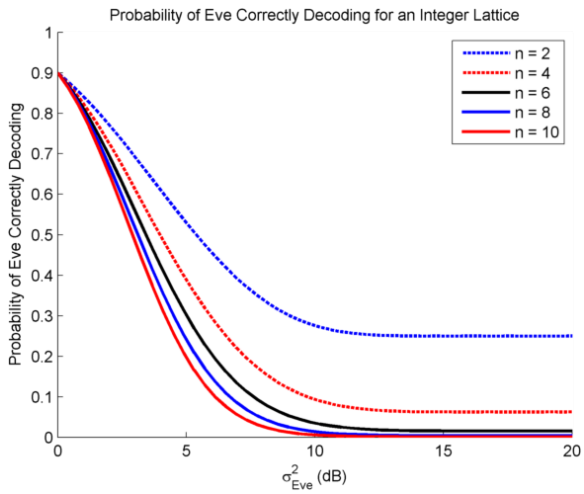


Figure 20. Probability of Eve Successfully Decoding with $\sigma^2_{Bob} = 1$

and is also related to the shortest vector problem. See [1, 17] as a starting point . While there exist algorithms that are fast, the ability of the algorithms to be implemented in a real communications system is currently questionable. The geometry of the decision regions in the integer lattice *greatly* simplifies matters. In the AWGN setting, the independence of the noise components means that each component can be handled independently and very fast algorithms exist for computing the modulo operation. Thus while the integer lattice is suboptimal, it's performance can be easily analyzed and it is a scheme that can probably be realized in hardware. In fact, in the original scope of this work, certain lattice systems were to be implemented and tested in hardware. The integer lattice is the only candidate class of lattices that this researcher has come across that seems viable for implementation. Given more time and resources, implementation of integer lattices in hardware is definitely a path that would be pursued.

The analysis of this section did not involve the use of a common random dither as described in Section 3. In practice, the use of this common randomness is required to get the benefit of the modulo channel, but from the viewpoint of the analysis of integer lattice it can be ignored. This is because in the worst-case scenario, which must be assumed in these types of eavesdropper analyzes, Eve also knows the common random dither. The common randomness can thus be ignored from the standpoint of analysis and focus can be given solely to the probability of Gaussian noise vectors lying in certain regions.

5 Newton's Method for Lattice Scaling

This section addresses the question of how to appropriately scale a lattice in order to meet the constraint that Bob correctly decodes the symbol with probability α. From a computational perspective, the performance of a lattice can be determined via Monte Carlo simulation. As was done in Section 4, a large number of Gaussian noise samples can be generated and the fraction that lies within the correct decoding region can be counted to estimate Eve's probability of correct decoding. Before this step however, the lattice must be appropriately scaled in order to guarantee Bob's constraint is met. In Section 4, the scaling factor β could be computed in closed form for the integer lattice by Equation 13 for two dimensions and more generally by Equation 17 for arbitrary n-dimensions. As was previously stated, the case where the scaling can be computed in closed form is rare. The reason it is possible for the integer lattice is due to the fact the decision regions have boundaries parallel to the coordinate axes and the use of the error function. This section will develop a Newton-type algorithm in order to compute the scaling of an arbitrary convex body, i.e. given a convex body A, what is the scaling factor β such that βA contains Gaussian probability mass equal to α. As in Section 4, an approximation will be made where it is assumed α is fairly large so that only a single decision region needs to be considered when computing β and all modulo-equivalent regions are ignored. With this assumption, because the Voronoi regions of a lattice are always convex, this Newton-type approach will enable the computation of the scaling factor for a lattice.

5.1 Equivalence of Scaling Integration Region and Variance

In this subsection it will be shown that when computing the Gaussian probability in a convex region, scaling the convex integration region is equivalent to scaling the variance of the Gauss measure (or associated random variable). This result will be illustrated via a one-dimensional example. For a zero mean Gaussian random variable with variance σ^2, the probability that the random variable lies in an interval $[a,b]$ is given by

$$\gamma([a,b]) = \int_a^b \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx. \tag{19}$$

Now suppose that the integration region is scaled by some positive number $\beta \in \mathbb{R}^{++}$. Then the Gauss measure in the interval $[\beta a, \beta b]$ is given by

$$\gamma([a\beta, b\beta]) = \int_{a\beta}^{b\beta} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right) dx. \tag{20}$$

This is equivalent to scaling the variance and can be seen by a change of variable: let $y = \frac{x}{\beta} \Rightarrow x = y\beta$ and thus $\beta dy = dx$ leading to

$$\gamma([a\beta, b\beta]) = \int_a^b \frac{\beta}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\beta^2 y^2}{2\sigma^2}\right) dx. \tag{21}$$

Define a new variance $\tilde{\sigma}^2 = \frac{\sigma^2}{\beta^2}$ and plug this into Equation 21 to get

$$\gamma([a,b]) = \int_a^b \frac{1}{\sqrt{2\pi\tilde{\sigma}^2}} \exp\left(-\frac{x^2}{2\tilde{\sigma}^2}\right) dx. \tag{22}$$

Equation 22 is identical to Equation 19 except for the new scaled variance, therefore in the one-dimensional case the equivalence has been established. This also holds in higher dimensions for a convex integration region using the same change of variable $\tilde{\sigma}^2 = \frac{\sigma^2}{\beta^2}$.

5.2 Scaling the One-Dimensional Lattice

As mentioned in the introduction of this section, it is assumed that α is large enough such that when computed the probability of Bob correctly decoding, focus can be given solely to the decision region corresponding to the transmitted code point and all modulo equivalent decision regions can be ignored. Thus for the one dimensional lattice, the single decision region is an interval, and without loss of generality the interval $[-1,1]$ can be considered. Let the variance of the Gaussian noise Bob experiences be given by σ^2. To meet Bob's probability constraint, the question is now what is the scaling factor β such that $\Pr[\mathbf{N} \in [-\beta,\beta]] = \alpha$ where $\mathbf{N} \sim N\left(0,\sigma^2\right)$ is the Gaussian noise random variable? This question is answered by setting the following equation equal to α

$$f(\beta) = \int_{-1}^{1} \frac{1}{\sqrt{2\pi\sigma^2/\beta^2}} \exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right) dx \tag{23}$$

i.e. $f(\beta) = \alpha$. A visualization of $f(\beta)$ is given in Figure 21. Note that $f(\beta)$ is a monotonically increasing function and $f : \mathbb{R}^{++} \to (0,1)$. In the figure the highlighted point is for $\beta = 1$, where the expected answer is roughly 0.68, the probability of a Gaussian random variable being within one standard deviation.

Newton's method is an iterative method to find solutions x of problems of the form $g(x) = 0$. For the problem under consideration, set $g(\beta) = f(\beta) - \alpha = 0$. The iterative update is given by

$$\beta^{[n+1]} = \beta^{[n]} - \frac{g\left(\beta^{[n]}\right)}{g'\left(\beta^{[n]}\right)} \tag{24}$$

where the superscript $[n]$ denotes the iteration number and g' denotes the derivative of g with respect to β. For details of Newton's method or its derivation, see [33]. To use Newton's method, the derivative $f'(\beta)$ has to be evaluated. In this one-dimensional case over the interval $[-1,1]$, the derivative is given by

$$\begin{aligned} \frac{d}{d\beta} f(\beta) &= \frac{\partial}{\partial\beta} \int_{-1}^{1} \frac{1}{\sqrt{2\pi\sigma^2/\beta^2}} \exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right) dx \\ &= \frac{\sqrt{\frac{2}{\pi}} \exp\left(-\frac{\beta^2}{2\sigma^2}\right)}{\beta\sqrt{\frac{\sigma^2}{\beta^2}}}. \end{aligned} \tag{25}$$

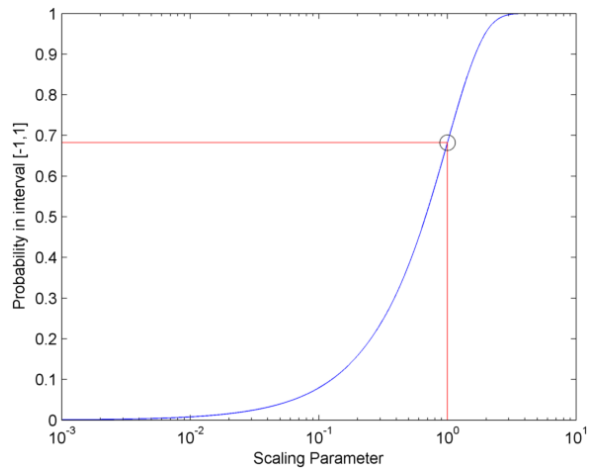


Figure 21. Gaussian Probability of Interval [-1,1] as a Function of β

For the one-dimensional case on the interval, the problem is now effectively solved since $f(\beta)$ and $f'(\beta)$ can be evaluated. Because f is monotonically increasing, Newton's algorithm will converge to the solution β such that $f(\beta) = \alpha$. This one dimensional example, while being illuminating, is too simple. Ultimately $f(\beta)$ and $f'(\beta)$ need to be evaluated on the Voronoi regions of arbitrary lattices and not just on the interval $[-1,1]$. Some additional insight can be gained by the one dimensional case before considering the general n-dimensional problem. Consider the partial derivative of the integrand $\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right)$:

$$\begin{aligned}\frac{\partial}{\partial\beta}\left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right)\right) &= \frac{(\sigma^2-\beta^2x^2)}{\sqrt{2\pi\sigma^2/\beta^2}\beta\sigma^2}\exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right)\\ &= \frac{1}{\beta\sqrt{2\pi\sigma^2/\beta^2}}\exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right)\\ &\quad -\frac{\beta}{\sigma^2}\frac{x^2}{\sqrt{2\pi\sigma^2/\beta^2}}\exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right). \qquad (26)\end{aligned}$$

Taking the integral of Equation 26 over the interval $[-1,1]$ yields Equation 25. To gain more insight, redefine $f(\beta)$, call it $\tilde{f}(\beta)$, to be the integral over an arbitrary interval A rather than the interval $[-1,1]$ and consider the integral of Equation 26 over the interval A

$$\begin{aligned}\int_A\frac{\partial}{\partial\beta}\left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right)\right)dx = &\frac{1}{\beta}\underbrace{\int_A\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right)dx}_{a_1}-\\ &\frac{\beta}{\sigma^2}\underbrace{\int_A\frac{x^2}{\sqrt{2\pi\sigma^2/\beta^2}}\exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right)dx}_{a_2}. \qquad (27)\end{aligned}$$

With the redefined $\tilde{f}(\beta)$, notice that the first term, a_1 in Equation 27 is equal to $\frac{1}{\beta}\tilde{f}(\beta)$. Looking at the second term a_2 in Equation 27, notice that if the integration region had been $\mathbb{R}$ rather than the interval A, the integral in term a_2 would evaluate to the variance $\frac{\sigma^2}{\beta^2}$. Thus, if the integration region A were replaced by $\mathbb{R}$ in term a_2, $\tilde{f}'(\beta)$ would evaluate to

$$\tilde{f}'(\beta) = \int_A \frac{\partial}{\partial\beta}\left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\exp\left(-\frac{x^2}{2\sigma^2/\beta^2}\right)\right)dx = \frac{1}{\beta}\tilde{f}(\beta) - \frac{\beta}{\sigma^2}\cdot\frac{\sigma^2}{\beta^2} = \frac{1}{\beta}\tilde{f}(\beta) - \frac{1}{\beta}. \quad (28)$$

The issue with the above formula is that the integration region is in fact A and not $\mathbb{R}$. The question now becomes how to evaluate the second term on the region A, or how to approximate this term. This leads to questions regarding the norm squared of truncated random variables.

5.3 N-Dimensional Lattice Scaling

In n-dimensions, the Gaussian probability of being in an n-dimensional set A with a scaling factor β is

$$f(\beta) = \int_A \left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\right)^n \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2/\beta^2}\right)d\mathbf{x}. \quad (29)$$

This is the n-dimensional generalization of Equation 23 over arbitrary sets A. Attention will be restricted to symmetric convex sets since A will be the Voronoi region of an n-dimensional lattice. As in the one-dimensional case, finding the lattice scaling β such that Bob's probability constraint is met amounts to finding the value of β such that $f(\beta) = \alpha$. Use of Newton's method to solve this problem amounts to computing $f(\beta)$ and $f'(\beta)$ as seen by Equation 24. The evaluation of these function over an arbitrary Voronoi region A cannot be computed in closed form, so a Monte Carlo approach will be taken.

Recall the equivalence between scaling the integration region or the variance:

$$f(\beta) = \int_A \left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\right)^n \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2/\beta^2}\right)d\mathbf{x} = \int_{\beta A} \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2}\right)d\mathbf{x}. \quad (30)$$

Generating a large number of Gaussian random variables, the value of $f(\beta)$ can be approximated by finding the proportion of the random sample found in βA or the proportion of the random sample scaled by $1/\beta$ that lies in A. The larger the number of samples, the better the approximation. For an arbitrary lattice it can be a difficult problem to compute the fraction of points lying in a Voronoi region A. The derivative $f'(\beta)$ must also be approximated. The derivative is given by the expression

$$\begin{aligned} f'(\beta) &= \frac{\partial}{\partial\beta}\int_A \left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\right)^n \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2/\beta^2}\right)d\mathbf{x} \\ &= \int_A \frac{\partial}{\partial\beta}\left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\right)^n \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2/\beta^2}\right)d\mathbf{x}. \end{aligned} \quad (31)$$

Evaluating the partial derivative of the integrand yields

$$\frac{\partial}{\partial\beta}\left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\right)^n \exp\left(-\frac{\|\mathbf{x}\|^2}{2\sigma^2/\beta^2}\right) = \frac{n}{\beta}\underbrace{\left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\right)^n \exp\left(\frac{\|\mathbf{x}\|^2}{2\sigma^2/\beta^2}\right)}_{b_1}$$
$$-\frac{\beta}{\sigma^2}\underbrace{\left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\right)^n \|\mathbf{x}\|^2 \exp\left(\frac{\|\mathbf{x}\|^2}{2\sigma^2/\beta^2}\right)}_{b_2}. \quad (32)$$

Equation 32 is the n-dimensional generalization of Equation 27. The integral of Equation 32 over the region A equals $f'(\beta)$. The integral of the term b_1 equals the original $f(\beta)$. The integral over the second term b_2 is equal to the expectation of the random variable $\mathbf{Y} = \|\mathbf{X}\cdot\mathbf{1}_A\|^2$, where $\mathbf{1}_A$ is the indicator function which is 1 when $\mathbf{X} \in A$ and 0 otherwise. This is the truncated random variable alluded to at the end of the previous section. To see this, notice

$$\int_A \left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\right)^n \|\mathbf{x}\|^2 \exp\left(\frac{\|\mathbf{x}\|^2}{2\sigma^2/\beta^2}\right) d\mathbf{x} = \int \left(\frac{1}{\sqrt{2\pi\sigma^2/\beta^2}}\right)^n \|\mathbf{x}\|^2 \exp\left(\frac{\|\mathbf{x}\|^2}{2\sigma^2/\beta^2}\right)\cdot \mathbf{1}_A d\mathbf{x}$$
$$= \mathbb{E}_{N'}\left[\|\mathbf{X}\cdot\mathbf{1}_A\|^2\right] \quad (33)$$

where the expectation is taken with respect to an n-dimensional Gaussian random variable with $\mathbf{0}$ mean and covariance matrix $\frac{\sigma^2}{\beta^2}\mathbf{I}$. Combining the integrals of terms b_1 and b_2 over the region A gives the equation

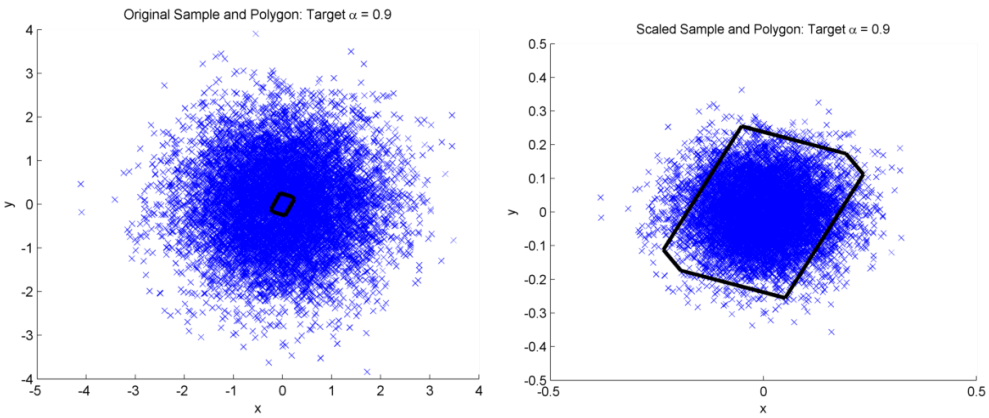
$$f'(t) = \frac{n}{\beta} f(\beta) - \frac{\beta}{\sigma^2}\mathbb{E}_{N'}\left[\|\mathbf{X}\cdot\mathbf{1}_A\|^2\right]. \quad (34)$$

As was previously discussed, $f(\beta)$ can be approximated via Monte Carlo methods. Equation 34 suggests that $f'(\beta)$ can also be approximated in this manner. The first term in $f'(\beta)$ equals a scaled version of $f(\beta)$ so can clearly be approximated in this way. To approximate the second term, generate (or re-use) a sample of n-dimensional Gaussian random variables with mean $\mathbf{0}$ and covariance $\frac{\sigma^2}{\beta^2}\mathbf{I}$. Find the Gaussian samples that lie in the set A, and then sum the norm-squared of those samples. Finally, divide this sum by the number of total samples (not just those lying in A) and the result approximates $\mathbb{E}_{N'}\left[\|\mathbf{X}\cdot\mathbf{1}_A\|^2\right]$. Having approximated all of the necessary terms, the Newton method can be implemented.

5.4 Notes on Computing Newton's Method

The ability to quickly compute the lattice scaling parameter β has made the computation analysis of lattices possible. Prior to adapting Newton's method to the lattice scaling parameter problem, a binary search type method was employed to determine β. Even in very simple cases, this procedure could take tens of minutes in MATLAB on a desktop computer. Newton's algorithm computes β with better accuracy in far less time, often less than a second in MATLAB on a

desktop computer for some problems. Figure 22 shows a lattice Voronoi region from a randomly generated Construction A lattice (to be discussed in detail in Section 6) and a sample of Gaussian random variables in blue. Figure 22(a) shows the original sample of Gaussian random variables drawn from $N(0,\mathbf{I})$ and the unscaled lattice. Figure 22(b) shows the scaled samples (which is equivalent to scaling the lattice). For this instance, Newton's algorithm converged in two iterations. Newton's method requires a tolerance parameter to know when to stop iterating, i.e. δ such that $|f(\beta)-\alpha| < \delta$. In this case $\delta = 10^{-4}$. Note that δ cannot be chosen too small because estimates of $f(\beta)$ and $f'(\beta)$ are used rather than the exact values so it may not be possible to make $|f(\beta)-\alpha| < \delta$ for δ too small. In higher dimensions even Newton's algorithm may slow down to the point that it becomes impractical. As most of the analysis in this work considered relatively small dimensions this was not an issue.



(a) Original Sample **(b)** Scaled Sample

Figure 22. Scaling Samples For Probability Constraint $\alpha = 0.90$

6 Random Lattices: Construction A

It is a very difficult problem to construct lattices that solve the desired optimization problem: minimize the probability that Eve decodes a symbol correctly given that Bob decodes correctly with probability α. Rather than try to optimize over the space of lattices, an alternative approach can be taken. Generate random lattices from a certain ensemble of lattices and compute some statistic of the performance over the ensemble. For example, given an ensemble of random lattices, it may be possible to compute the average performance of the system. The question now becomes how to find a good lattice ensemble. The lattice ensemble that will be analyzed in this section is known as "Construction A" and has proven useful in more tradition communication problems. Erez and Zamir solved a long standing problem in [16] when they showed that the capacity of the AWGN channel can be achieved using lattices drawn from the construction A ensemble. The same authors and Litsyn show in [15] that construction A lattices are also useful in solving other types of problems such as quantization, sphere packing and sphere coverings. It is the hope that construction A lattices yield good results for the secure communication problem. The analysis in this section contains partial results but does not fully answer this question. When approaching the secure communication problem using Construction A lattices, the original goal is to find the best possible lattice in the ensemble. This, not surprising, turns out to be a difficult problem. The results in this section will be computational. Several desirable properties of construction A lattices and their distributions will be shown for a range of low dimensional lattices. Connections will be made with the "smoothing parameter" of a lattice introduced in the cryptography community. Lastly initial investigations into the use of semidefinite programming to find the best construction A lattices will be described.

6.1 Construction A Lattices

This discussion of construction A lattices is from [15]. More details on these lattices can also be found in [12, 29]. The ensemble of construction A lattice is parameterized by three values: n,p, and k. The parameter n defines the dimension of the lattice, p has to be a prime number and is related to the code rate of the lattice and $k \leq n$ is also related to the rate of the lattice. Construction A lattices are generated as follows:

1. Pick a prime number p.
2. Create a $n \times k$ lattice generator matrix G by drawing each element of G i.i.d. uniformly from the set $\{0, 1, \ldots, p-1\}$, i.e. $G_{i,j} \sim \text{Unif}(0, \ldots, p-1)$ for $i = 1, \ldots, n$ and $j = 1, \ldots, k$.
3. Define the set of code points as $C = \left\{\mathbf{x} = G\mathbf{y} \bmod p : \mathbf{y} \in \mathbb{Z}_p^k\right\}$ where $\mathbf{y} \in \mathbb{Z}_p^k$ is a vector whose components are integers between 0 and $p-1$ and where the modulo operation is taken component-wise.
4. Define a new set of code points Λ^* by dividing C by p, i.e. dividing each code point by p and $\Lambda^* = \frac{1}{p}C$. Everyone code point in Λ^* thus lies in the the hypercube $[0,1]^n$.

5. The construction A lattice is given by $\Lambda = C^* + \mathbb{Z}^n$, i.e. Λ^* is copied and offset by all possible integer lattice points and is thus replicated over $\mathbb{R}^n$.

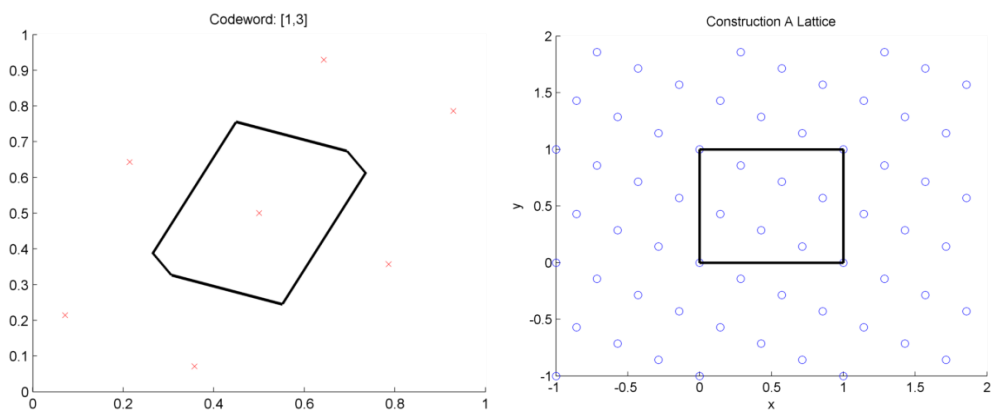
This is fairly abstract, so considering the following example. Let $n = 2$, $p = 7$, and $k = 1$. Suppose that G is randomly drawn and the realization is $G = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$. $\mathbf{y} \in \mathbb{Z}^1$, so the possible values of $\mathbf{y}$ are $\{0,1,2,3,4,5,6\}$. From Step 3 of the procedure, the set of code points is equal to

$$C = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \begin{bmatrix} 2 \\ 6 \end{bmatrix}, \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \begin{bmatrix} 5 \\ 1 \end{bmatrix}, \begin{bmatrix} 6 \\ 4 \end{bmatrix} \right\}. \tag{35}$$

Step 4 of the procedure normalizes all the code points so that they lie in $[0,1]^2$

$$\Lambda^* = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \frac{1}{7} \\ \frac{3}{7} \end{bmatrix}, \begin{bmatrix} \frac{2}{7} \\ \frac{6}{7} \end{bmatrix}, \begin{bmatrix} \frac{3}{7} \\ \frac{2}{7} \end{bmatrix}, \begin{bmatrix} \frac{4}{7} \\ \frac{5}{7} \end{bmatrix}, \begin{bmatrix} \frac{5}{7} \\ \frac{1}{7} \end{bmatrix}, \begin{bmatrix} \frac{6}{7} \\ \frac{4}{7} \end{bmatrix} \right\}. \tag{36}$$

Figure 23(a) shows the set of points Λ^* and the Voronoi region of a single point. Step 5 replicates Λ^* and is illustrated in Figure 23(b), where the black box shows the original set Λ^*. It is known



(a) Scaled Code Points Λ^* **(b)** Replication of Λ^*

Figure 23. Example of a Construction A Lattice for $n = 2$, $p = 7$ and $k = 1$

that this procedure will generate a lattice [12]. An important property of construction A lattices is that with high probability $|\Lambda^*| = p^k$. This property connects p and k to the rate of the lattice. If a construction A lattice is used for communication then there are $M = p^k$ code points that can be sent (with high probability). A lattice fails to have p^k code points when the randomly generated G is nonsingular, but as long as n and k are chosen careful the probability of this event is small. In the analysis of this work, as well as other ([16]), it will be assumed $k = 1$. In this case, $M = p$ and the probability that G is nonsingular is p^{-n}, the probability that G is the all-zeros vector. A nice feature of Construction A lattices is that the sub-lattice with which the modulo operation is taken

with respect to is the integer lattice. For an arbitrary lattice and sub-lattice the modulo operation can be very difficult to compute. In the case of Construction A lattices the modulo operation is the same as with the integer lattice, which is very easy to compute, namely each component can be handled individually. Most of the computational power is required to determine which code point within the hypercube is closest to the received code point after the modulo operation has been performed. Therefore among general lattices, construction A lattices are more practical from the perspective of implementation.

6.2 Properties of Construction A Lattices

The original goal of using construction A lattices was to find the best lattice among the ensemble. By the results of Section 2, best was determined to mean the lattice with the maximum minimum distance, or the lattice whose Voronoi regions contain a sphere with the largest radius. This turns out to be a difficult problem. For a specific triplet of n,k,p, in theory all of the construction A lattices can be enumerated. Given a complete enumeration, the best lattice (the lattice with the maximum minimum distance), can be selected. As the parameters increase however, the number of lattices grows and an complete enumeration may not be possible. It is known [15, 16, 29] that as the parameters grow, the performance increases. It is this property that is used in theoretical analysis. From a "practical" perspective however, n,k,p must be fixed (non-asymptotic) with complexity issues in mind. It is this reason focus is given to trying to find the best lattice for a fixed set of parameters. In this section, focus will be given to the case $n = 2$ and $k = 1$. In this case, the set $\mathbf{y} = \{0,\ldots,p-1\}$ and G is just a randomly generated two-dimensional vector.

Equivalent Lattices

When constructing lattices for $n = 2, k = 1$ and a fixed p, it becomes apparent that some generators G, although not identical, lead to the same performance. The first type of equivalence is algebraic in nature. Consider again the set C in Equation 35 from the example in the previous section. The generator matrix was $G = \begin{bmatrix}1 & 3\end{bmatrix}^T$. Had G been any of the other non-zero vectors in C, i.e. $\begin{bmatrix}2 & 6\end{bmatrix}^T$, $\begin{bmatrix}3 & 2\end{bmatrix}^T$, etc., the same lattice would be generated. In algebraic terminology, the code point vectors in the set C belong to an orbit and are a subgroup. In the above example, there are $p^2 = 49$ possible G vectors, thus the probability of randomly generating the lattice shown in Figure 23(b) is $\frac{6}{49}$. This result is from the fact that if G is any of the six non-zero vectors in C, the lattice will be generated, so six out of the possible 49 vectors will generate the lattice. Following this type of analysis, the number of possible lattices is given by $\frac{p^2-1}{p-1}$ since there are a total of p^2-1 possible vectors for G and $p-1$ of those vectors lead to the same lattice.

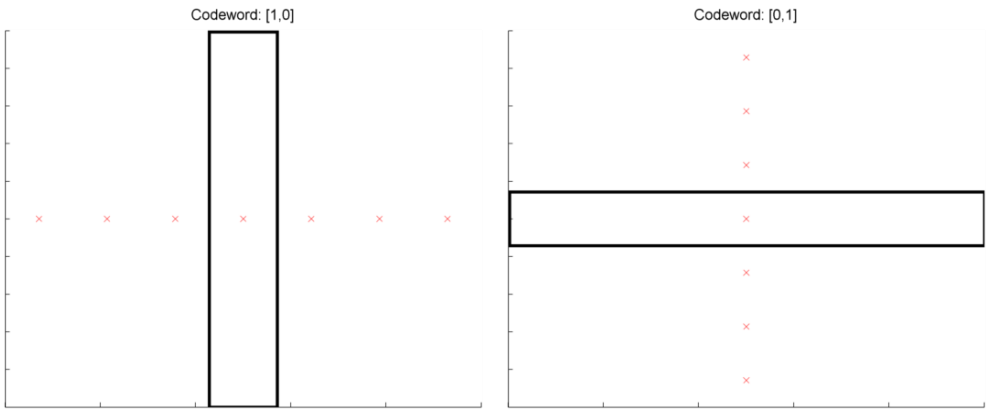
The second type of equivalence is geometric in nature. As an extreme case, consider the lattice generated by $G_1 = \begin{bmatrix}1 & 0\end{bmatrix}^T$ so that the set C_1 is given by

$$C_1 = \left\{ \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}2\\0\end{bmatrix}, \begin{bmatrix}3\\0\end{bmatrix}, \begin{bmatrix}4\\0\end{bmatrix}, \begin{bmatrix}5\\0\end{bmatrix}, \begin{bmatrix}6\\0\end{bmatrix} \right\} \tag{37}$$

as shown in Figure 24(a) and consider the lattice generated by $G_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}^T$ so that

$$C_2 = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 0 \\ 4 \end{bmatrix}, \begin{bmatrix} 0 \\ 5 \end{bmatrix}, \begin{bmatrix} 0 \\ 6 \end{bmatrix} \right\} \tag{38}$$

as shown in Figure 24(a). While these lattices are not formally equivalent, they have equivalent performance when used in an AWGN channel. The reason for this is that the decision regions are congruent, differing only by a rotation, e.g. in Figure 24, the decision region in the right hand figure is the decision region of the left hand figure rotated by 90 degrees. In the AWGN channel, the



(a) Code Points C_1 and Decision Region **(b)** Code Points C_2 and Decision Region

Figure 24. Lattices with the same Performance but Different Generators

Gaussian noise is rotation invariant so the Gaussian probability of lying in the decision region when centered on a code point is the same in both figures. Two lattices which have congruent decision lattices will be called geometrically equivalent. After accounting for algebraically equivalent lattices, the number of unique lattices has a nice closed form solution. This does not seem to be the case for the number when considering geometrically equivalent lattices. All geometrically equivalent lattice will be said to belong to an equivalence class. Figure 25 shows the number of geometric equivalence classes for $n = 2, k = 1$ and a function of the prime p. The plot looks roughly linear, but is erratic in the sense that no closed form solution for this plot seems to be known. The number of geometric equivalence classes is interesting because this number, in this example, appears to grow linearly in p, where as the number of possible lattices grows as p^2 (in the two dimensional case).

Maximum Minimum Distance

One of the primary lattice parameters of interest is the maximum minimum distance. As indicated in Section 2, the more "spherical" the decision region, the better the performance. Figure 26

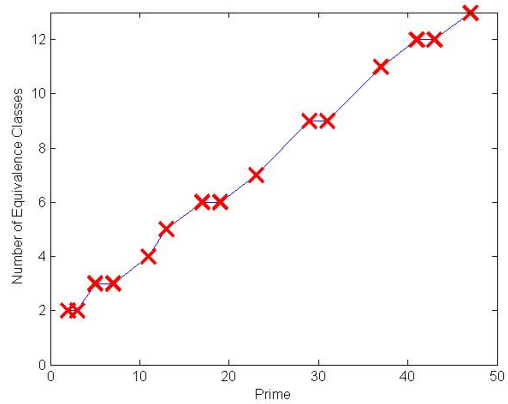
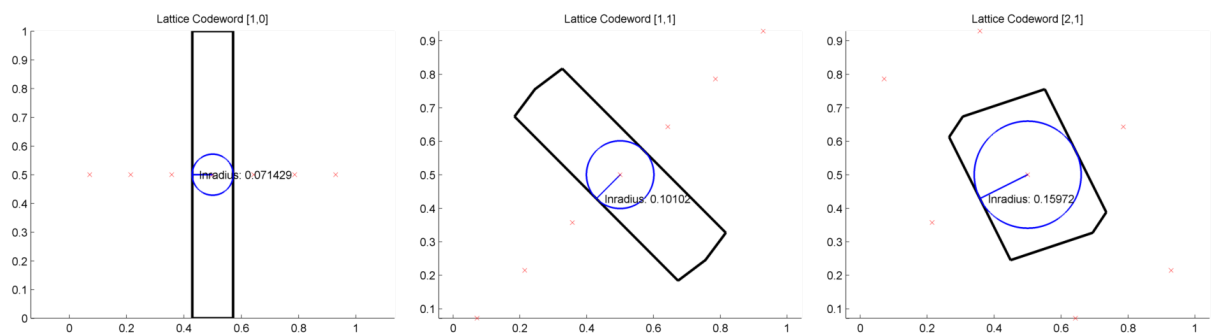


Figure 25. Number of Equivalence Classes for $n = 2$, $k = 1$ as a Function of prime p

shows an example of a construction A lattice from each of the three geometric equivalence classes for $p = 7$. Each lattice is also plotted with the circle of maximum radius that can fit inside the decision region. Figure 27 shows the ability of Eve to correctly decode for of each geometric



(a) **(b)** **(c)**

Figure 26. Unique Construction A Lattices for $n = 2$, $k = 1$, $p = 7$

equivalence class for this example assuming the Bob's probability constraint is $\alpha = 0.90$. As expected, the lattice in Figure 26(c) performs the best (decays to the probability of random guessing the fastest) since it has the largest minimum distance of all possible construction A lattices with these parameters. Conversely, the lattice in Figure 26(a) has the worst performance and also has the smallest minimum distance. For large values of the parameters n, k and p, it may be impossible to enumerate all possible lattices and select the best lattice. It would be ideal to be able to compute the probability of generating specific minimum distances for a given set of parameters. The explicit

computation of these problems is also a very challenging problem, but it can be simulated to try to gain some intuition. The behavior of the minimum distance distribution seems to be fairly erratic.

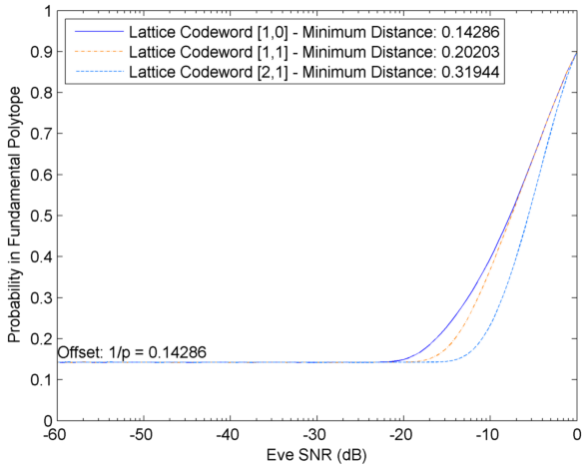
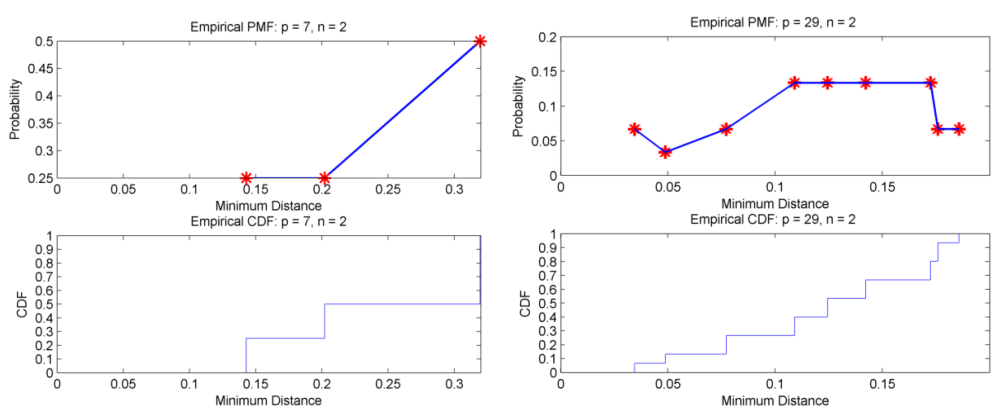


Figure 27. Probability of Eve Correctly Decoding for Each Geometric Equivalence Class: $n = 2, k = 1, p = 7$

For example, in Figure 28(a), the parameters are $n = 2, k = 1$ and $p = 7$ and there are three unique minimum distances (corresponding to the red stars) which relate to the three unique geometric equivalence classes as has been seen. This situation is rather fortuitous since the largest minimum



(a) p = 7 **(b)** p = 29

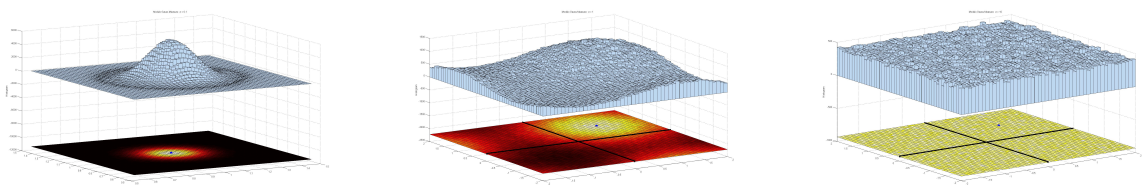
Figure 28. Minimum Distance Distributions

distance is also the most probable; there is a 50% chance that a randomly generated construction A lattice will have the best possible minimum distance. The situation in Figure 28(b) is quite different. In this case, the parameters are $n = 2, k = 1$ and $p = 29$ and notice there there are a total

of nine geometric equivalence classes and there are four minimum distances which are equally and maximally probable, but are not the maximum minimum distance. In fact, the maximum minimum distance is generated with a probability equal to $\frac{1}{15}$. Much computational effort has been expended in order to characterize these distributions and compute the maximum minimum distances for a range of parameters. From a computational perspective, much of the code was written in python, and even C-optimized python called cython, in order to speed up the computations as much as possible. Given more time and resources, it would be interesting to compute the average performance of construction A lattices for a fixed set of parameters since explicitly characterizing the maximum minimum distance or its distribution has proven to be exceptionally difficult.

Smoothing Parameter

Another possible avenue of attack is via the connection between the performance of a lattice in terms of inhibiting Eve's ability to decode and the "smoothing parameter" introduced in the computer science realm [32, 37]. Informally, the smoothing parameter is the smallest variance such that the modulo-Gaussian measure appears uniform. This is the best case scenario in terms of trying to confuse eavesdroppers. Thus, from the stand point of secure communications, lattices with the smallest smoothing parameters have the best performance. Figure 29 shows the effects



(a) **(b)** **(c)**

Figure 29. Modulo-Gauss Measure

of the variance on the modulo-Gauss measure. Figure 29(a) shows a two dimensional Gaussian probability density centered on a code point with no boundary or modulo effects. The upper plot is a histogram representation of the density and the lower plot is a heat map representation. Figure 29(b) shows a Gaussian distribution with larger variance centered on a QAM code point and with the modulo effects from wrapping around the edges. Notices that the histogram and heat map look more uniform as compared with Figure 29(a). In Figure 29(c), the variance is selected to be larger than the smoothing parameter. In this case, the modulo-Gauss measure looks uniform. In this case, the probability of correctly decoding the transmitted code point is equal to the probability of random guessing. Making a connection with Figure 27, the smoothing parameter corresponds to the point where the probability plateaus, which is the probability of random guessing. From Figure 27, the lattice with the largest minimum distance plateuas first corresponding to the best performance and the largest smoothing parameter.

For completeness the the smoothing parameter will be formally defined, but the intuition of the smoothing parameter given above should not be lost. First, the concept of a dual lattice must be defined. Let Λ be a lattice, then the dual lattice Λ^* is defined as

$$\Lambda^* = \{\mathbf{x} : \forall \mathbf{y} \in \Lambda \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z}\} \tag{39}$$

where $\langle \mathbf{x}, \mathbf{y} \rangle$ is the standard inner product. In words, the dual lattice is the set of vectors in $\mathbb{R}^n$ whose inner product with every lattice vector $\mathbf{y} \in \Lambda$ is an integer. The simplest example of a dual lattice is the integer lattice whose dual lattice is itself. The smoothing parameter can now be defined ([32]): For an n-dimensional lattice Λ, and a positive real number $\varepsilon > 0$, the lattice's smoothing parameter $\eta_\varepsilon(\Lambda)$ is the smallest s such that $\rho_{1/s}(\Lambda^* \backslash \{\mathbf{0}\})$. Here $\rho_{1/s}$ is a Gauss measure with variance $1/s$ (see [32] for details), and $\Lambda^* \backslash \{\mathbf{0}\}$ is the dual lattice without the origin. This definition is pretty abstract, but luckily [32] proved some easy to interpret bounds on the smoothing parameter. The first bound is given as follows: for any n-dimensional lattice Λ, $\eta_\varepsilon(\Lambda) \leq \sqrt{n}/\lambda_1(\Lambda^*)$, where $\lambda_1(\Lambda^*)$ denotes the shortest distance between two points in the dual lattice. The second bound is given as follows: For an n-dimensional lattice Λ and $\varepsilon > 0$,

$$\eta_\varepsilon(\Lambda) \leq \sqrt{\frac{\log(2n(1+1/\varepsilon))}{\pi}} \cdot \lambda_n(\Lambda) \tag{40}$$

where $\lambda_n(\Lambda)$ is the nth successive minima (see [32] for details).

The idea is now to find the best lattice, or specifically construction A lattices, in terms of smoothing parameter. While the smoothing parameter itself may be difficult to compute, it may be possible to optimize one of the two bounds just given. In this way the problem is converted from the problem of computing the maximum minimum distance to computing the maximum minimum distance of the dual lattice for the first bound, or finding the nth successive minima of the lattice in the second bound. These problems seem to be as difficult as the original problem, but perhaps more energy could be spent investigating this connection.

6.3 Semidefinite Programming and Construction A Lattice Optimization

Semidefinite programming (SDP) is a generalization of linear programming and the optimization variables are real symmetric matrices. Let S^n denote the set of symmetric $n \times n$ matrices and S^n_+ is the set of positive semidefinite matrices. For $X, Y \in S^n$, define the inner product as $\langle X, Y \rangle \triangleq \operatorname{trace} XY$. A semidefinite program is then given by the following optimization problem:

$$\begin{aligned} \underset{X}{\text{minimize}} \quad & \langle X, Y \rangle \\ \text{subject to} \quad & \langle A_i, X \rangle = b_i \\ & X \succeq 0 \end{aligned}$$

where $X \in S^n$ is the optimization variable $b \in \mathbb{R}^m$ and $C, A_i \in S^n$. $X \succeq 0$ is a common notation for $X \in S^n_+$. The theory of semidefinite programming will not be discussed at length here. For more information consult [28, 45, 48].

For this section, redefine C as a $n \times p$ matrix whose columns are code points in a construction A lattice. From the example given by Equation 35, C will now be a matrix containing those code points as columns. Now define the matrix P as the matrix C augmented with $2^n - 1$ vectors in $\mathbb{R}^n$ containing all possible combinations of 0 and p except the all zeros vector which is already included, e.g. for the example given in Equation 35

$$C = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 0 & 7 & 7 \\ 0 & 3 & 6 & 2 & 5 & 1 & 4 & 7 & 0 & 7 \end{bmatrix}. \tag{41}$$

where the last three columns are the augmented vectors. Now define the matrix $Y = C^T C \in \mathbb{R}^{(p+2^n-1)\times(p+2^n-1)}$. Y is an example of a Gram matrix for the points defined in the columns of C. Define the matrix D as a matrix whose elements represent the pairwise distance between points. The matrix D can be computed from a list of points C via the Gram matrix Y:

$$D = \text{diag}(Y)\mathbf{1}^T + \mathbf{1}\text{diag}(Y)^T - 2Y \tag{42}$$

where $\text{diag}(Y)$ is an n-dimensional vector of the diagonal elements of Y and $\mathbf{1}$ is an n-dimensional vector of ones. There is also a pseudo-inverse $K^\dagger$ that maps a pairwise distance matrix D to a Gram matrix Y:

$$K^\dagger(D) = -\frac{1}{2} J \text{offDiag}(D) J \tag{43}$$

where $J \triangleq \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ and $\text{offDiag}(D)$ is a matrix equal to D except the diagonal elements are set to zero. The pairwise distance matrix D is an element in what is known as the cone of Euclidean distance matrices (EDM). More information about the EDM cone can be found in [14, 25]. The relationship with the secure communications problem is that the minimum off-diagonal entry of D generated from C is (essentially) the minimum distance of the lattice. A few tricks must be performed to guarantee that it is the minimum distance of the lattice.

The minimum distance of a construction A lattice is related to the minimum off diagonal value of the pairwise distance matrix D, so an optimal construction A lattice can be constructed by optimizing the the minimum off-diagonal element of D under the constraint that the result be a valid construction A lattice. The set of constraints for the smallest parameters n, k, p can be effectively enumerated. The semidefinite program to maximize the minimum distance of a construction A lattice takes several tricks in order to function properly even in the simplest cases. Figure 30 shows the result of a semidefinite program designed to optimize the minimum distance of a construction A lattice for the parameters $n = 2, k = 1, p = 5$. The resulting lattice is correct; it belongs to the geometric equivalence class with the largest minimum distance. The code used to generate Figure 30 can be found in Appendix A. The code is written in MATLAB and uses the sedumi SDP solver [42] and the YALMIP interpreter [30]. Unfortunately this code does not extend beyond the simplest cases. It is unclear what the exact issue is, but is most likely due to the constraints. For larger parameter values, the constraints used in this example are not adequate and lead to incorrect solutions. Beyond the incorrect solutions, the computation effort grows dramatically with parameter values and as it stands is not a viable method for even modest problems. Even so, with more time and effort a more formalized set of constraints could possibly be found that appropriately captures the construction A lattice constraints and is computationally feasible.

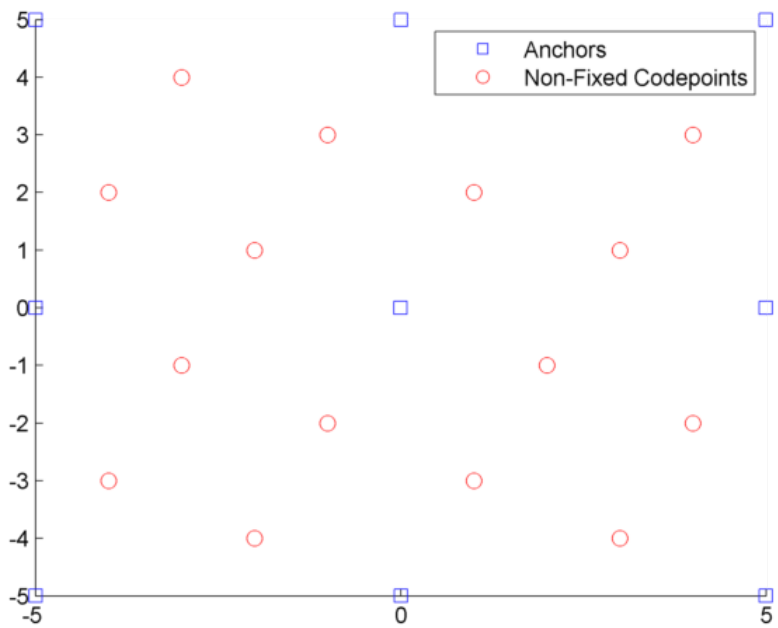


Figure 30. SDP Lattice Output for $n = 2, p = 5$

7 Optimization of Lattices

Towards the very end of this research project, a few branches of research were discovered that seems to hold promise for generating lattices with favorable properties, at least for dimensions $n < 20$. One branch is that of the geometry of positive definite quadratic forms [40, 41]. Utilizing results from this theory and the theory of random walks, Andreanov and Scardicchio [2] in the statistical mechanics community were able to generate very dense lattices. In fact, they were able to recover many of the densest known lattices in certain dimensions with high probability. Marcotte and Torquato [31] linearized the objective function of lattice density and applied a sequence of linear programs as in [46] to yield dense lattices. Again, they were able to generate some of the densest known lattices in certain dimensions with this technique. Other techniques have been developed in the statistical mechanics literature, e.g. [24], for generating lattices, but currently [2, 31] seem to represent the cutting edge in terms of performance and accuracy. These results are also very recent, having been published only in the past few months. In this section some of these techniques will be briefly described. The analysis of these techniques as applied to the secure communications problem would be very interesting, and would be the direction pursued had there been more time.

7.1 Positive Definite Quadratic Forms and Lattices

Basics of Positive Quadratic Forms

This section briefly describes the relationship between lattices and positive definite quadratic forms (PQF). The results are taken from [40], which should be consulted for more details. As defined in [40], a quadratic form $Q : V \to K$ is a homogeneous polynomial of degree 2 defined for a field K and a vector space V over K. For the the work related to lattice, the field is generally either $\mathbb{R}$ or $\mathbb{Q}$, i.e. the reals or the rationals. This definition is very abstract, but it can be reformulated in terms of matrices. All real quadratic forms in n variables can be identified with the space of real symmetric $n \times n$ matrices $S^n = \{Q \in \mathbb{R}^{n\times n} | Q^t = Q\}$. A quadratic form is then given by the equation $\mathbf{x}^t Q\mathbf{x}$, where $\mathbf{x} \in \mathbb{R}^n$. For example, the homogeneous polynomial of degree two $x_1^2 + x_1x_2 + x_2^2$ is associated with the matrix $Q = \begin{bmatrix} 1 & \frac{1}{2} \\ \frac{1}{2} & 1 \end{bmatrix}$. A quadratic form is positive definite if $\mathbf{x}^t Q\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^n \backslash 0$. The matrices Q associated with positive definite quadratic forms are precisely the set of positive definite matrices $S^n_>$.

A fundamental question in the theory of quadratic forms is given a vector $\mathbf{x} \in \mathbb{Z}^n$ (i.e. all the coordinates are integer valued), when does $\mathbf{x}^t Q\mathbf{x} = \alpha$ when $\alpha \in \mathbb{N}$. It turns out that if the quadratic form is positive definite, not only does a solution exist, but there exists a smallest natural number α for which $\mathbf{x}^t Q\mathbf{x}$ has an integral solution. This minimum is called the arithmetical minimum and is denotes

$$\lambda(Q) = \min_{\mathbf{x}\in\mathbb{Z}^n\backslash\{0\}} \mathbf{x}^t Q\mathbf{x}$$

Going back to the previous example, the equation $x_1^2 + x_1x_2 + x_2^2$ is actually a positive definite

quadratic form because the associated Q matrix is positive definite. The arithmetical minimum of of this PQF is 1. This can be seen geometrically in Figure 31. Drawing the ellipse $x_1^2 + x_1x_2 + x_2^2 \leq 1$, the ellipse intersects the integer lattice at six points, and at the six points the function evaluates to 1. Denote the points of the integer lattice that achieve the arithmetical minimum

$$\text{Min}Q = \left\{ \mathbf{x} \in \mathbb{Z}^d | \mathbf{x}^t Q \mathbf{x} = \lambda(Q) \right\}.$$

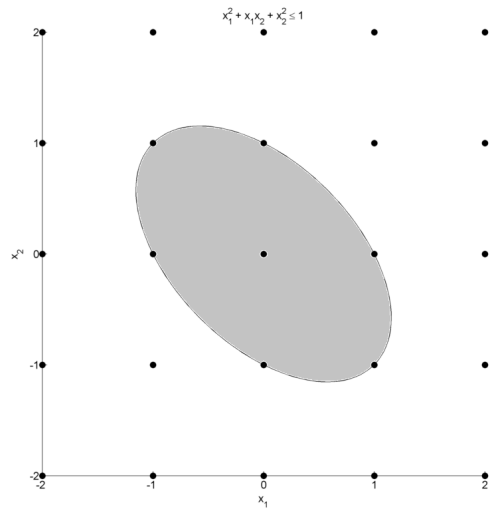


Figure 31. Ellipse from the equation $x_1^2 + x_1x_2 + x_2^2$

Relationship with Lattices

Having described the very basics of positive definite quadratic forms, the connection with lattices will now be given. PQFs are given by positive definite matrices Q, so a Cholesky decomposition of Q can be performed: $Q = A^t A$, where A is a member of the general linear group

$$GL_n(\mathbb{R}) = \left\{ a \in \mathbb{R}^{n \times n} | \det A \neq 0 \right\}.$$

Given the Cholesky decomposition of Q, the PQF can be rewritten as

$$\mathbf{x}^t Q \mathbf{x} = \mathbf{x}^t A^t A \mathbf{x} = \|A\mathbf{x}\|^2$$

which is the squared length of the vector $A\mathbf{x}$. Therefore, when $\mathbf{x}$ is restricted to be an integer vector, as was considered in the previous section, the matrix A can be viewed as the generator matrix of a lattice. With this relationship established, many lattice problems can be viewed in the framework of PQFs.

The lattice sphere packing problem places a sphere of radius R at each lattice point such that the sphere do not overlap. The largest value of R is the packing radius of the lattice. The packing

radius of a lattice can be expressed in terms of the arithmetical minimum: let A be a generator matrix of a lattice Λ and $Q = A^t A$, then the packing radius of Λ is given by

$$\lambda(\Lambda) = \frac{\sqrt{\lambda(Q)}}{2}.$$

This makes sense, as $\lambda(Q)$ represents the shortest distance squared between two points in a lattice given by A, and thus the largest possible radius such that spheres do not intersect is given by half the distance. The sphere packing density of a lattice Λ is given by the fraction of space occupied by the spheres centered on the lattice points with radius $\lambda(\Lambda)$ and is given by the equation

$$\delta(\Lambda) = \frac{\mathrm{vol}\left(\lambda(\Lambda)B^n\right)}{\det\Lambda} = \frac{\lambda(\Lambda)^n \mathrm{vol}B^n}{\det\Lambda} = \left(\frac{\lambda(Q)}{(\det Q)^{1/n}}\right)^{n/2} \cdot \frac{\mathrm{vol}B^n}{2^n}$$

where $\mathrm{vol}B^n$ is the volume of an n-dimensional unit sphere.

A common lattice problem is to try to find a lattice Λ that has the maximum density $\delta(\Lambda)$ in a given dimension. A key property of $\delta(\Lambda)$ is that it is invariant to lattice isometries (e.g. rotation) and scaling. Therefore, if the lattice density is to be maximized, a constraint on $\lambda(Q)$ or equivalently on $\lambda(\Lambda)$ can be enforced and then maximizing $\delta(\Lambda)$ is equivalent to minimize $\det\Lambda$ or $\det Q$. This constraint optimization problem is a way of rephrasing the maximum lattice density problem in terms of PQFs, and is just as difficult to solve, but this formulation has led to very successful algorithms (at least in dimensions $n < 20$) as will be discussed later. The lattice sphere covering problem ask the question: if a sphere is placed at every lattice point, how big must the radius be such that every point in $\mathbb{R}^n$ is inside a sphere? The smallest such radius is called the covering radius of a lattice. Formally, the covering radius $\mu(\Lambda)$ of lattice Λ is defined as:

$$\mu(\Lambda) = \inf\{\mu > 0 | \Lambda + \mu B^n = \mathbb{R}^n\}$$

where the addition $\Lambda + \mu B^n = \mathbb{R}^n$ is a Minkowski sum. Analogous to lattice sphere packing density, the lattice sphere covering density is given by

$$\Theta(\Lambda) = \mu(\Lambda)^n \mathrm{vol}B^n \mathrm{dens}\Lambda$$

where $\mathrm{dens}\Lambda$ is the density of the lattice. To optimize the sphere covering, the lattice with the smallest possible covering radius is desired:

$$\Theta_n = \inf\{\Theta(\Lambda) | \Lambda \text{ an n-dimensional lattice}\}.$$

As with the sphere packing problem, this problem only has known solutions in a few cases. The last classic lattice problem addressed in this section is the lattice sphere packing-covering problem. This problem is equivalent to the problem of trying to maximize the minimum distance between lattice points, which is a very interesting problem from the secure communications perspective as indicated by the analysis of Section 6. For a lattice Λ, the packing-covering constant is given by $\gamma(\Lambda) = \mu(\Lambda)/\lambda(\Lambda)$. Maximizing the minimum distance corresponds to minimizing $\gamma(\Lambda)$. The optimal lattice in n dimensions is given by the solution to

$$\gamma_n = \inf\{\gamma(\Lambda) | \Lambda \text{ an n-dimensional lattice}\}.$$

Algorithm for Lattice Sphere Packings

In this section Voronoi's algorithm will be described, which in theory can find the lattice Λ with greatest sphere packing density $\delta(\Lambda)$ for an arbitrary dimension n. In practice, the algorithm only works when the dimension n is very small because the complexity quickly grows with dimension. The first step towards developing this algorithm was a fact previously mentioned: the lattice sphere packing density δ is invariant to scaling and therefore attention can be restricted to lattices (or PQFs) with a fixed arithmetical minimum λ. With λ fixed, the task is then to minimize the determinant of Q. More generally, one can consider all PQFs Q with arithmetical minimum greater than or equal to λ (the constraint matter only in that $\lambda(Q) \not< \lambda$). Define the set of PQFs that satisfy this generalized constraint as

$$P_\lambda = \{Q \in S^n_> | \lambda(Q) \geq \lambda\}.$$

The task of maximizing the lattice sphere packing density is thus equivalent to minimizing the determinant of PQFs $Q \in P_\lambda$ for an arbitrary positive λ. The set P_λ is called the Ryshkov polyhedron. With out going into the details (see [40]), it turns out that the lattice achieving the greatest sphere packing density has to be a vertex of the Ryshkov polyhedron. The vertices of the Ryshkov polyhedron are called perfect forms.

The Voronoi algorithm takes as input a dimension n and outputs a list of perfect forms. Once the list is generated, the sphere packing density of each lattice in the list can be computed and the lattice with the largest density is guaranteed to be the best lattice. The details of the algorithm will not be presented here, suffice to say that it can be computed by a computer. The issue is that the number of vertices (or perfect forms) grows super-exponentially. In 8 dimensions there are 10,916 perfect forms and in dimension 9 there are over 500,000 and the total number is unknown. Thus up to 8 dimensions, the optimal sphere packing lattice is known, but beyond 9 dimensions the optimal lattice is not known since not all perfect forms are known. This algorithm can still be useful in that it can generate dense lattices but there is no guarantee about optimality without enumeration of all the perfect forms. Using the algorithm in an exploratory mode such as this reaches a limit around dimension 20 as indicated in [2, 31].

Algorithms for Coverings and Packing-Coverings

The details of the algorithms designed to find good lattice coverings and lattice packing-coverings will not be covered in detail. The interested reader can consult Chapter 5 in [40]. The algorithms can be obtained from Schürmann's website [39] and use as subroutines the MAXDET algorithm [51] and the lrs package [3]. Using this algorithm, Schürmann is able to generate the best known lattice coverings up to dimension 15 and packing-coverings in dimension 7. The algorithms presented in Chapter 5 of [40], as with Voronoi's algorithm described in previous section, have computational issues that arise very quickly in even moderate dimensions, hence only results up to dimension 15. Given more time however, it would be interesting to implement this algorithm to generate good lattice packing-coverings and test their performance in the secure communications problem.

7.2 Statistical Physics Approach

The statistical physics community has recently addressed the problem of developing algorithms that generate dense sphere packings. The two algorithms that will be focused on in this section are [2, 31], both published very recently. The algorithm of [2] is based on the Ryshkov polyhedron described in the previous section. Rather than attempt to enumerate all of the perfect forms as in the Voronoi algorithm, the vertices are transversed via a random walk. Their algorithm is thus a randomized version of the Voronoi algorithm. In dimensions $n \leq 7$ this algorithm finds all the densest known lattices (Note: the densest lattice are only known for dimensions 1 through 8 and dimension 24). This is because the number of vertices in the Ryshkov polyhedron is very small up to dimension 7, so the probability of hitting the densest lattice with this randomized algorithm is very high. As the dimension increases, the probability of the random walk hitting the densest lattice in a reasonable running time decreases. For dimensions $n \geq 20$ the algorithm is ineffective.

The algorithm used in [31] is a refinement and specialization of an algorithm developed in [46] for the lattice sphere packing problem. Let A be the generator matrix of an n-dimensional lattice Λ. At every iteration of the algorithm, the generator matrix is updated by the equation

$$A \rightarrow A + \varepsilon A$$

where $\varepsilon \in \mathbb{R}^{n \times n}$ is an $n \times n$ symmetric strain matrix. The question then becomes how to select the strain matrix ε so that the sequence of generator matrices converges to a dense lattice. The authors linearize the non-linear objective function of lattice sphere packing density in terms of the strain matrix, and derive constraints for ε. The choice of ε under the linearized objective and derived constraints results in a linear program which can readily be solved using a computer. Each iteration of the results in a generator matrix of a denser lattice and eventually the converges to a local maximum. For low dimensions, the algorithm was able to converge to the global densest lattice. As with the previous algorithm as the dimension is increased, the ability of the algorithm to find the global maximum is decreased. The authors report that for $n = 19$, the algorithm converged to the global optimum only 0.009% of the time. Regardless, this algorithm is fast and has yields the best performance to date.

7.3 Applications to Secure Communications

The ability to generate quality lattices or quantify the quality of random lattices is a challenging problem. In Section 6, Construction A lattices were considered because they have been used successfully in the analysis of communication systems and are easy to generate. An interesting alternative to the use of Construction A lattices is to use the algorithms described in this section to generate lattices. These algorithms have led to state of the art results in terms of finding some of the best known lattices for the sphere packing, sphere covering, and packing-covering problems. Had there been more time, it definitely would be interesting to use the algorithms to generate lattices for the secure communications problem.

This page is intentionally left blank.

8 Conclusion

This LDRD has been very challenging. Many lattice problem are notoriously difficult and it has been difficult to make headway in any particular direction, which is why so many techniques and procedure have been attempting during the course of this project. With that having been said, modest result have been obtained in several different directions.

In Section 2, the focus is given to a single decision region. It is then shown that decision regions that are "more spherical" perform better in terms of confusing a potential eavesdropper. While a single decision region does not take into account the whole lattice, it is a very good approximation when the eavesdropper's noise variance is close to that of the intended user's noise variance.

Section 3 discusses how common randomness can be used to convert the additive noise channel into the modulo-Λ additive noise channel. This is important for the potential implementation of a system. Original coset coding considered the use of an entire lattice for confusing eavesdroppers. In this case, there are an infinite number of code points and requires an infinite amount of power. A proposed solution in the literature is to truncate the lattice after a certain point so that the resulting constellation has finite power, but has a loss of performance. Conversion to the modulo-Λ additive noise channel has finite power and no loss in performance. This connection seems to not have been mentioned in the secure communications community.

The integer lattice is the simplest lattice to analyze and also possibly implement. Section 4 provides the analysis of these lattices. The probability constraint that Bob successfully decode with probability α is approximated by the probability of a Gaussian random variable being in the decision region of the transmitted code point - no cosets are considered. This approximation is very accurate for α near unity. With this approximation, the exact scaling of the integer lattice can be found to meet the probability constraint in any dimension by the use of the error function. Given the exact scaling factor, the eavesdropper's performance is computed via simulation. It is noted that integer lattices are the easiest lattice to implement because the modulo operation can be taken on each dimension independently.

Bob is required to correctly decode with probability α. Given an arbitrary lattice, computing the scaling factor to so that this constraint is met is challenging. Section 5 discusses the adaption of Newton's method in order to quickly compute this scaling factor. The proposed algorithm is an approximation as it requires estimates of several quantities which cannot be computed explicitly. Computer experiments show that the approximate algorithm works very well in practice.

Section 6 discusses the use of construction A lattices. This random ensemble of lattices has been used in previous communication settings to great theoretical success. In this research, their application has been limited by the lack of understanding of critical parameters, namely the maximum minimum distance. Simulations are run to try to gain insight into the maximum minimum distance, however the behavior is erratic. Semidefinite programs are used to try to find the optimal construction A lattice for a fixed set of parameters. The result of these optimization algorithms were only successful for the simplest cases.

Section 7 describes an approach to lattice optimization from the view point of positive definite quadratic forms and statistical mechanics. Algorithms have been developed in these communities that have shown great success in finding lattices with very good parameters. The intuition behind these ideas is described.

Future Research

Given more time and resources, there are several possible routes that could lead to new results. One possible route is an attempt to increase simulation accuracy and Monte Carlo parameter estimation. For example, Newton's method in Section 5 requires the generation of i.i.d. Gaussian samples to estimate the probabilities of certain sets and expectations of certain random variables. A new sampling technique called "herding" has been introduced in [49, 50] that has shown convergence rates faster than that of i.i.d. sampling. The use of these algorithms could increase accuracy and speed since fewer samples may be required. These algorithms could also be used to increase the accuracy of the analysis of lattices, such as the integer lattice analysis of Section 4

Another possible route would be the Monte Carlo simulation of construction A lattices. The results found in Section 6 were focused on finding optimal lattices from the ensemble. As was mentioned there, this avenue of attack proved difficult. Rather than attempting to find the best possible construction A lattice for a fixed set of parameters, a large sample of construction A lattice could be drawn and statistics on the performance of the ensemble could be estimated, such as the average performance and variance.

Section 7 gives several algorithms that have been used successful in other scientific domains for finding good lattices. Some of these results are very new and discovered by this author very recently. It would be interesting to implement and run these algorithms to attempt to find lattices that are good for secure communications. The most efficient algorithm in the statistical physics community ([31]) is based on the iterative use of linear programs. Over the past decade, semidefinite programming has been used to improve the results given by linear programs in some programs ([4, 10, 11, 36, 38]). It would be interesting to see if semidefinite programming could be used to improve the results of the iterative linear programming approach in [31].

This research focused on the AWGN channel model. Originally, additional channel models were to be considered, but were not due to difficulties that arise even in the simple AWGN channel model. Another route for future research could involve adapting the procedures described here to analyze other channel models, such as fading and MIMO models, from a computational perspective.

References

[1] Erik Agrell, Thomas Eriksson, Alexander Vardy, and Kenneth Zeger. Closest point search in lattices. *Information Theory, IEEE Transactions on*, 48(8):2201–2214, 2002.

[2] A. Andreanov and A. Scardicchio. Random perfect lattices and the sphere packing problem. *Phys. Rev. E*, 86:041117, Oct 2012.

[3] D Avis. lrs homepage, 2005.

[4] Christine Bachoc and Frank Vallentin. New upper bounds for kissing numbers from semidefinite programming. *Journal of the American Mathematical Society*, 21(3):909–924, 2008.

[5] Alexander Barvinok. Math 710: Measure concentration. *Lecture Notes*, 2005.

[6] J-C Belfiore and Frédérique Oggier. Secrecy gain: a wiretap lattice code design. In *Information Theory and its Applications (ISITA), 2010 International Symposium on*, pages 174–178. IEEE, 2010.

[7] J-C Belfiore and Frédérique Oggier. Lattice code design for the rayleigh fading wiretap channel. In *Communications Workshops (ICC), 2011 IEEE International Conference on*, pages 1–5. IEEE, 2011.

[8] J-C Belfiore and Patrick Solé. Unimodular lattices for the gaussian wiretap channel. In *Information Theory Workshop (ITW), 2010 IEEE*, pages 1–5. IEEE, 2010.

[9] Vladimir Igorevič Bogačev. *Gaussian measures*. Number 62. AMS Bookstore, 1998.

[10] Henry Cohn and Noam Elkies. New upper bounds on sphere packings i. *Annals of Mathematics*, pages 689–714, 2003.

[11] Henry Cohn and Abhinav Kumar. Universally optimal distribution of points on spheres. *Journal of the American Mathematical Society*, 20(1):99–148, 2007.

[12] J.H. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*. Springer, 2010.

[13] D Cordero-Erausquin, M Fradelizi, and B Maurey. The (b) conjecture for the gaussian measure of dilates of symmetric convex sets and related problems. *Journal of Functional Analysis*, 214(2):410–427, 2004.

[14] Jon Dattorro. *Convex optimization and Euclidean distance geometry*. Meboo Publishing USA, 2005.

[15] Uri Erez, Simon Litsyn, and Ram Zamir. Lattices which are good for (almost) everything. *Information Theory, IEEE Transactions on*, 51(10):3401–3416, 2005.

[16] Uri Erez and Ram Zamir. Achieving 1/2 log (1 + snr) on the awgn channel with lattice encoding and decoding. *IEEE Transaction on Information Theory*, 50(10):2293 – 2314, October 2004.

[17] Ulrich Fincke and Michael Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of computation*, 44(170):463–471, 1985.

[18] G David Forney Jr. Coset codes. i. introduction and geometrical classification. *Information Theory, IEEE Transactions on*, 34(5):1123–1151, 1988.

[19] G David Forney Jr. Coset codes. ii. binary lattices and related codes. *Information Theory, IEEE Transactions on*, 34(5):1152–1187, 1988.

[20] G David Forney Jr. Geometrically uniform codes. *Information Theory, IEEE Transactions on*, 37(5):1241–1260, 1991.

[21] Richard Gardner and Artem Zvavitch. Gaussian brunn-minkowski inequalities. *Transactions of the American Mathematical Society*, 362(10):5333–5353, 2010.

[22] Thomas C Hales. A proof of the kepler conjecture. *The Annals of Mathematics*, 162(3):1065–1185, 2005.

[23] Yoav Kallus. Statistical mechanics of the lattice sphere packing problem. *Phys. Rev. E*, 87:063307, Jun 2013.

[24] Yoav Kallus, Veit Elser, and Simon Gravel. Method for dense packing discovery. *Physical Review E*, 82(5):056707, 2010.

[25] Nathan Krislock and Henry Wolkowicz. *Euclidean distance matrices and applications*. Springer, 2012.

[26] Rafał Latała. On some inequalities for gaussian measures. *arXiv preprint math/0304343*, 2003.

[27] Rafał Latała and Krzysztof Oleszkiewicz. Gaussian measures of dilatations of convex symmetric sets. *The Annals of Probability*, 27(4):1922–1938, 1999.

[28] Monique Laurent and Franz Rendl. Semidefinite programming and integer programming. *Handbooks in Operations Research and Management Science*, 12:393–514, 2005.

[29] H-A Loeliger. Averaging bounds for lattices and linear codes. *Information Theory, IEEE Transactions on*, 43(6):1767–1773, 1997.

[30] Johan Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004.

[31] Étienne Marcotte and Salvatore Torquato. Efficient linear programming algorithm to generate the densest lattice sphere packings. *Phys. Rev. E*, 87:063303, Jun 2013.

[32] Daniele Micciancio and Oded Regev. Worst-case to average-case reductions based on gaussian measures. *SIAM Journal on Computing*, 37(1):267–302, 2007.

[33] Todd K Moon and Wynn C Stirling. *Mathematical methods and algorithms for signal processing*, volume 1. Prentice hall Upper Saddle River, 2000.

[34] Frédérique Oggier, Patrick Solé, and Jean-Claude Belfiore. Lattice codes for the wiretap gaussian channel: Construction and analysis. *arXiv preprint arXiv:1103.4086*, 2011.

[35] Giorgio Parisi. On the most compact regular lattices in large dimensions: A statistical mechanical approach. *Journal of Statistical Physics*, 132(2):207–234, 2008.

[36] Florian Pfender and Günter M Ziegler. Kissing numbers, sphere packings, and some unexpected proofs. 2004.

[37] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[38] Alexander Schrijver. New code upper bounds from the terwilliger algebra and semidefinite programming. *Information Theory, IEEE Transactions on*, 51(8):2859–2866, 2005.

[39] A Schürmann and F Vallentin. Geometry of lattices and algorithms, 2005.

[40] Achill Schurmann. *Computational Geometry of Positive Definite Quadratic Forms: Polyhedral Reduction Theories, Algorithms and Applications*, volume 48. AMS Bookstore, 2009.

[41] Achill Schurmann and Frank Vallentin. Computational approaches to lattice packing and covering problems. *Discrete & Computational Geometry*, 35(1):73–116, 2006.

[42] Jos F Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.

[43] Michel Talagrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications Mathématiques de l'Institut des Hautes Etudes Scientifiques*, 81(1):73–205, 1995.

[44] Michel Talagrand. New concentration inequalities in product spaces. *Inventiones mathematicae*, 126(3):505–563, 1996.

[45] Michael J Todd. Semidefinite optimization. *Acta Numerica*, 10(515-560):126, 2001.

[46] S. Torquato and Y. Jiao. Robust algorithm to generate a diverse class of dense disordered and ordered sphere packings via linear programming. *Phys. Rev. E*, 82:061302, Dec 2010.

[47] S. Torquato and F. H. Stillinger. Jammed hard-particle packings: From kepler to bernal and beyond. *Rev. Mod. Phys.*, 82:2633–2672, Sep 2010.

[48] Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1):49–95, 1996.

[49] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1121–1128. ACM, 2009.

[50] Max Welling and Yutian Chen. Statistical inference using weak chaos and infinite memory. In *Journal of Physics: Conference Series*, volume 233, page 012005. IOP Publishing, 2010.

[51] SP Wu, L Vandenberghe, and S Boyd. Maxdet: Software for determinant maximization problems. *Information Systems Laboratory, Stanford University*, 1996.

[52] Aaron D Wyner. The wire-tap channel. *Bell Syst. Tech. J.*, 54:1355–1387, 1975.

A Semidefinite Program to Optimize Construction A Lattices

```
% Using semidefinite programming to find optimal two-dimensional ...
    construction A lattice
% The dimension will be fixed to N = 2
% The prime seed will be fixed at p = 5
% This is the largest case in which the constraints led to a known ...
    optimal construction A solution
%
% Using sedumi and YALPMIP interpreter
% Have linear constraints and "addivitity" constraint

DIM = 2;     % Dimension = 2
prime = 5;   % Prime seed p = 5
% For N = 2, number of points = 4*(prime-1) + 1 (origin) + 8 (other ...
    anchors)
numPOINTS = 4*(prime-1) + 1 + 8;
numANCHORS = 9; % number of anchors = 9 for N = 2

%Constant Constraints
%---------------------------------------------
%---------------------------------------------
% Anchor points
A = prime*[0 0;1 0;1 1;0 1;-1 1;-1 0;-1 -1;0 -1;1 -1];
% Gram Matrix of Anchor Points
Y1 = A*A';
Yconstraint = A*A';
% Pairwise Distance Matrix D of anchor points
Dconstraint = diag(Yconstraint)*ones(numANCHORS,1)' + ...
    ones(numANCHORS,1)*diag(Yconstraint)' - 2*Yconstraint;

% Vector of all ones
e = ones(numPOINTS,1);
J = eye(numPOINTS) - (1/numPOINTS)*e*e';

% Optimization Variables
Y = sdpvar(numPOINTS,numPOINTS);
X = sdpvar(DIM,numPOINTS);

% Pairwise Distance Matrix D Optimization Variable
D = diag(Y)*e' + e*diag(Y)' - 2*Y;
% Pseudo-inverse Optimization Variable
Kinv = -0.5*J*(D - diag(diag(D)))*J;

% Constraint on Gram Matrix, both equality and PSD
Dconstraint1 = [Kinv≥0,Y(1:numANCHORS,1:numANCHORS) == Yconstraint,Y≥0];

% New Constraint from Equation (970) of Dattorro
Eq970 = [eye(DIM) X;X' Y];
Dconstraint1 = Dconstraint1 + set(Eq970 ≥ 0);
% Add equality constraints on X a la 970
Dconstraint1 = Dconstraint1 + set(X(:,1:numANCHORS) == A');
```

```
%-----------------------------------------------
%-----------------------------------------------

% Make Diagonal Constraints
%-----------------------------------------------
%-----------------------------------------------
% For N = 2, first non-anchor point occurs at index 10
ind1 = 10:1:(10+prime-2);
ind2 = ind1 + 1*(prime-1)*ones(1,length(ind1));
Dconstraint2 = set(D(ind1,ind1) == D(ind2,ind2));
% 4 Regions
for i = 2:3
    ind2 = ind1 + i*(prime-1)*ones(1,length(ind1));
    Dconstraint2 = Dconstraint2 + set(D(ind1,ind1) == D(ind2,ind2));
end

% Inter-codeword Distance
%-----------------------------------------------
%-----------------------------------------------

%Make sure inter-codepoint distance isn't too big
Dtemp = D(ind1,ind1);
% Dconstraint2 = Dconstraint2 + set(Dtemp(:) ≤ DIM*prime^2);
Dconstraint2 = Dconstraint2 + set(D(ind1,ind1) ≤ DIM*prime^2);
%-----------------------------------------------
%-----------------------------------------------


% Distance Constraint
%-----------------------------------------------
%-----------------------------------------------

Dconstraint2 = Dconstraint2 + set(D(1,:) ≤ DIM*prime^2);

% %-----------------------------------------------
% %-----------------------------------------------

% Make Shifted Points Constraints
%-----------------------------------------------
%-----------------------------------------------

% Generalized for arbitrary prime
% Four sets of indices for for regions (since N = 2)
ind2 = ind1 + 1*(prime-1)*ones(1,length(ind1));
ind3 = ind1 + 2*(prime-1)*ones(1,length(ind1));
ind4 = ind1 + 3*(prime-1)*ones(1,length(ind1));

% Start with Origin
%-----------------------------------------------

% Looking up to the right
Dconstraint3 = [D(1,ind1) == D(6,ind2)];
Dconstraint3 = Dconstraint3 + set(D(1,ind1) == D(7,ind3));
```

```
Dconstraint3 = Dconstraint3 + set(D(1,ind1) == D(8,ind4));

% Looking down to the left
Dconstraint3 = Dconstraint3 + set(D(1,ind3) == D(3,ind1));
Dconstraint3 = Dconstraint3 + set(D(1,ind3) == D(4,ind2));
Dconstraint3 = Dconstraint3 + set(D(1,ind3) == D(2,ind4));

% Looking up to the left
Dconstraint3 = Dconstraint3 + set(D(1,ind2) == D(2,ind1));
Dconstraint3 = Dconstraint3 + set(D(1,ind2) == D(8,ind3));
Dconstraint3 = Dconstraint3 + set(D(1,ind2) == D(9,ind4));

% Looking Down to the right
Dconstraint3 = Dconstraint3 + set(D(1,ind4) == D(4,ind1));
Dconstraint3 = Dconstraint3 + set(D(1,ind4) == D(5,ind2));
Dconstraint3 = Dconstraint3 + set(D(1,ind4) == D(6,ind3));

% %-----------------------------------------------

% Translation of Codepoints
%-----------------------------------------------

% Generalize for different prime
Dconstraint4 = [diag(D(ind1,ind2)) == prime^2*ones(prime-1,1)];
% Dconstraint4 = [diag(D(ind1,ind2)) == prime^2*ones(2,1)];
Dconstraint4 = Dconstraint4 + set(diag(D(ind1,ind4)) == ...
    prime^2*ones(prime-1,1));
% Dconstraint4 = Dconstraint4 + set(diag(D(ind1,ind4)) == ...
    prime^2*ones(2,1));
Dconstraint4 = Dconstraint4 + set(diag(D(ind1,ind3)) == ...
    2*prime^2*ones(prime-1,1));
% Dconstraint4 = Dconstraint4 + set(diag(D(ind1,ind3)) == ...
    2*prime^2*ones(2,1));

%-----------------------------------------------

% Symmetry Constraint
%-----------------------------------------------
ind3_reverse = fliplr(ind3);
Dconstraint1 = Dconstraint1 + set(X(:,ind1) == -X(:,ind3_reverse));

% Sum Constraint
%-----------------------------------------------
Dconstraint1 = Dconstraint1 + set(X(:,ind1(1)) + X(:,ind4(1)) == ...
    X(:,ind1(2)));

%-----------------------------------------------
%-----------------------------------------------

% Create Total Constraints
%-----------------------------------------------
%-----------------------------------------------

```

```
TotalConstraints = Dconstraint1 + Dconstraint2 + Dconstraint3 + ...
    Dconstraint4;
% Adding minimum distance constraint?
Dtemp2 = D + DIM*eye(numPOINTS);
TotalConstraints = TotalConstraints + set(Dtemp2(:) ≥ DIM);

%-----------------------------------------------
%-----------------------------------------------

% Optimization Algorithm
%-----------------------------------------------
%-----------------------------------------------
% ops = sdpsettings('solver','sdpt3','verbose',1);
ops = sdpsettings('solver','SEDUMI','verbose',1);
iterCOUNTER = 1;
epsilon = 1e-10;
% maxITER = 5;
maxITER = 20;

%Regularization parameter
w = 1;

%Initial W
W = zeros(numPOINTS);
% W = zeros(6);
%Initial Kinv_rank
Kinv_rank = numPOINTS;
% Kinv_sol_eig_store = zeros(numPOINTS,maxITER);
Y_eig_store = zeros(numPOINTS,maxITER);
Kinv_rank_store = zeros(1,maxITER);
t_store = zeros(1,maxITER);
Xstore = zeros(2,numPOINTS,maxITER);
SecondTermStore = zeros(1,maxITER);

%Optimization Variables
%Ideally t would be an integer variable
t = sdpvar(1,1);
% t = intvar(1,1);
Z = D - t*(ones(numPOINTS) - diag(diag(ones(numPOINTS))));
TotalConstraints = TotalConstraints + set(Z(:) ≥ 0) + set(t≥0);

% Objective3 = -t;
% sol3 = solvesdp(TotalConstraints,Objective3,ops);

%Iterative Procedure
tic;
while(iterCOUNTER ≤ maxITER)
% while(iterCOUNTER < maxITER && Kinv_rank > DIM)

    %-----------------------------------------------
    % Objective1 = -t + w*trace(Kinv'*W);
    Objective1 = -t + w*trace(Y'*W);

    sol = solvesdp(TotalConstraints,Objective1,ops);
```

```
    % D_sol = double(D(1:8,1:8));
    % Kinv_sol = double(Kinv);
    Ysol = double(Y);
    t2 = double(t);
    %-----------------------------------------------

    %Second Optimization
    %-----------------------------------------------
    Wopt = sdpvar(numPOINTS,numPOINTS);
    Objective2 = -t2 + w*trace(Ysol'*Wopt);
    Constraints2 = [eye(numPOINTS)≥Wopt,Wopt≥0,trace(Wopt) == ...
        numPOINTS-DIM];
    sol2 = solvesdp(Constraints2,Objective2,ops);
    W = double(Wopt);
    %-----------------------------------------------

    Y_sol_eig = real(eig(Ysol));
    Y_eig_store(:,iterCOUNTER) = Y_sol_eig;
    Xstore(:,:,iterCOUNTER) = double(X);
    SecondTermStore(iterCOUNTER) = w*trace(Ysol'*W);
    t_store(iterCOUNTER) = double(t);

    %Iterate Counter
    iterCOUNTER = iterCOUNTER + 1;

end
toc;

% Conversion into numerical data types
Dsol = double(D);
Ysol = double(Y);
tsol = double(t);
Kinv_sol = double(Kinv);
Kinv_sol_eig = real(eig(Kinv_sol));
Xsol = double(X);

%---------------------------------------------
%---------------------------------------------


% Plot Resulting Lattice
%---------------------------------------------
figure;
scatter(Xsol(1,1:9),Xsol(2,1:9),50,'bs');
hold on;
scatter(Xsol(1,10:end),Xsol(2,10:end),50,'ro');
hold off;
legend('Anchors','Non-Fixed Codepoints');
```

DISTRIBUTION:

1	MS 9110	Jerry Brewer, 8229
1	MS 9110	Jay Brink, 8229
1	MS 9110	Tina Chou, 8229
1	MS 9110	Mike Foley, 8229
1	MS 9110	Jacob Goodwin, 8229
1	MS 9110	Jacques Kvam, 8229
1	MS 9110	Ratish Punnoose, 8229
1	MS 9110	Jamie Stamps, 8229
1	MS 9110	Kai Tsang, 8229
1	MS 9110	Ken Wallace, 8229
1	MS 9110	Peter Yang, 8229
1	MS 9110	Paul Yoon, 8229
1	MS 9153	Tim Shepodd, 8220
1	MS 9154	Bill Ballard, 8200
1	MS 0899	Technical Library, 8944 (electronic copy)
1	MS 0359	D. Chavez, LDRD Office, 1911

Sandia National Laboratories