Technische Universität München
Professur für Coding and Cryptography
Prof. Dr.-Ing. Antonia Wachter-Zeh

Master's Thesis

Decoding Problems via Quantization

Vorgelegt von:
Bharath Purtipli

München, Mai 2025

Betreut von:
Sebastian Bitzer, Emma Munisamy, Stefan Ritterhoff

Master's Thesis an der
Professur für Coding and Cryptography (COD)
der Technischen Universität München (TUM)
Titel : Decoding Problems via Quantization
Autor : Bharath Purtipli

Bharath Purtipli

Ich versichere hiermit wahrheitsgemäß, die Arbeit bis auf die dem Aufgabensteller bereits bekannte Hilfe selbständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderung entnommen wurde.

München, 22.05.2025

Ort, Datum (Bharath Purtipli)

Contents

Abstract **1**

1 Introduction **3**

2 Preliminaries **5**

2.1 Notation 5

2.2 Codes 5

2.2.1 Basics 6

2.2.2 Some Important Codes 7

2.2.3 Code-based Quantization 8

2.2.4 Hard Problems on Codes 9

2.3 Lattices 10

2.3.1 Basics 10

2.3.2 Lattice-based Quantization 11

2.3.3 Hard Problems on Lattices 11

2.4 Cryptography 12

2.4.1 Symmetric Cryptography 12

2.4.2 Asymmetric Cryptography 13

2.4.3 Post-Quantum Cryptography 13

2.5 Lossy Source Coding 14

3 Symmetric Cryptosystems **17**

3.1 Quantized Lattice-based Symmetric Cryptosystem 17

3.1.1 A Simple Lattice-based Symmetric Cryptosystem 17

3.1.2 Applying Quantization 18

3.1.3 Dithering 20

3.2 Quantized Code-based Symmetric Cryptosystem 20

3.2.1 Correctness of Decryption 21

3.3 Results 23

4 Asymmetric Cryptosystems **27**
4.1 Hamming Quasi-Cyclic (HQC) Cryptosystem 27
4.1.1 The Scheme 27
4.1.2 Correctness and Error Model 28
4.1.3 Decryption Failure Rate 29
4.2 Quantized HQC 31
4.2.1 Our Contribution 31
4.2.2 Correctness of Decryption 31
4.2.3 Methodology of Parameter Search 32
4.3 Results 34
4.3.1 Overview of the Search Results 34
4.3.2 Interpretation and Impact Analysis 35

5 Conclusion **37**

Bibliography **41**

List of Figures

3.1 An example of a two-dimensional lattice in $\mathbb{R}^2$ is shown here in blue dots. The elements of the cryptosystem are shown as vectors. 18

3.2 The quantization is performed with another two-dimensional lattice Q represented with grey dots. The effect of quantization is to map every vector $\mathbf{b}$ to the nearest lattice point from Q represented by $\lfloor \mathbf{b} \rceil_Q$. This also introduces an error, which is shown here in red $\{\mathbf{b}\}_Q$. 19

3.3 The effective channel "seen" by the code E at the decryption stage. 23

3.4 Asymptotic plaintext-to-ciphertext rate of the quantized (green) and non-quantized (blue) code-based symmetric cryptosystem. 24

3.5 Difference in plaintext-to-ciphertext rate of the quantized and non-quantized code-based symmetric cryptosystem with compression factor $\rho = 1 - H_2(p) + \delta_\rho$ for various δ_ρ. As the ρ moves away from the rate-distortion limit, the region of (τ, p) pairs that produce $\alpha > \alpha'$ shrinks. 26

4.1 Search strategy to find optimal Reed-Muller – Reed-Solomon (RMRS) code for quantized HQC. 33

List of Tables

3.1 Lattice-based symmetric cryptosystem by [MS23]. 17
3.2 Lattice-based symmetric cryptosystem with quantization by [MS23]. . . . 18
3.3 Lattice-based symmetric cryptosystem with quantization and dithering by [MS23]. 20
3.4 The novel code-based symmetric cryptosystem with quantization. 21
3.5 The novel code-based symmetric cryptosystem with dithering. The dithering vector $\mathbf{v}$ removes dependence between $\mathbf{e}$ and the quantization error $\{\mathbf{b}\}_Q$ 22

4.1 HQC Cryptosystem 28
4.2 Quantized HQC cryptosystem with dithering. 31
4.3 Optimal parameters chosen for the quantized HQC cryptosystem along with the computed tolerable Bernoulli error rate p and its components p^* from the intentional error and p_Q from quantization. 34
4.4 Comparison of ciphertext lengths between the current HQC standard and the quantized version obtained by the optimal parameters in Table 4.3 using the asymptotic rate-distortion bound Equation (2.3). 34
4.5 Comparison of ciphertext lengths between the current HQC standard and the quantized version obtained by the optimal parameters in Table 4.3 for the finite-length rate-distortion bound Equation (2.4). 35
4.6 Comparison of public-key lengths between the current HQC standard and the quantized version obtained by the optimal parameters in Table 4.3. . 35

Abstract

Due to the vulnerability of current state-of-the-art public-key cryptosystems to efficient attacks using quantum computers, novel quantum-resistant algorithms are required. Code-based cryptosystems such as Hamming Quasi-Cyclic (HQC) are of particular interest not only because they are assumed to be safe from quantum attacks, but they are also possible candidates for fully homomorphic encryption schemes [bacr25]. However, they suffer from large ciphertext sizes which leads to inefficient communication. In this work, we explore the feasibility of quantization as a means to reduce the ciphertext length in code-based cryptosystems. We successfully reduced the ciphertext length of the HQC scheme by 12% by introducing a trade-off in the computational complexity of encryption and decryption.

1 Introduction

Quantum computers pose a significant threat to the security of currently deployed public-key cryptographic infrastructure. Although state-of-the-art quantum computers are currently incapable of breaking the current standard public-key encryption, in the future they may be sufficiently powerful to retrospectively attack current communications. Algorithms like Shor's [Sho99] can efficiently break widely used systems such as Rivest, Shamir, Adleman (RSA) and Elliptic-Curve Cryptography (ECC) on a powerful enough quantum computer, necessitating the development and standardization of quantum-resistant alternatives. This field, known as Post Quantum Cryptography (PQC), explores various mathematical foundations, including lattice-based, multivariate, and code-based cryptography, to build new secure primitives.

Among these, code-based cryptosystems, pioneered by McEliece [McE78], have a long history and are valued for their potential for fast encryption and decryption operations. Their security often relies on the presumed hardness of decoding a random linear code, a problem considered NP-hard [BMvT78]. However, a persistent challenge for many code-based schemes is the considerable size of their public keys and, particularly, their ciphertexts. While various techniques exist to reduce key sizes, such as leveraging algebraic structures or amortization, ciphertext length remains a significant concern for practical deployment.

This thesis investigates the application of quantization, a technique derived from lossy source coding, as a means to reduce ciphertext lengths in code-based cryptosystems. Quantization aims to compress data by allowing a controlled amount of distortion, thereby trading perfect recovery for improved communication efficiency. The core idea is to represent the ciphertext, or a component thereof, not exactly, but by its closest point in a predefined, sparser set (a "quantization code" or lattice). Then, a more compact representation of this quantized ciphertext is transmitted, thereby saving communication costs. This approach has been explored in lattice-based cryptography [MS23], [LLL24] and offers a promising avenue for code-based schemes.

First, a generic symmetric code-based cryptosystem is examined to establish the fundamental principles and trade-offs involved. Subsequently, these concepts are applied to the HQC cryptosystem [AMAB$^+$25], a promising candidate in the National Insti-

tute for Standards and Technology (NIST) PQC standardization process. We develop a methodology for integrating quantization into HQC, analyze its impact on parameters, and evaluate the potential ciphertext reduction. A systematic search through several possible parameter sets and codes was done to find the optimal scheme with the shortest possible ciphertext. A reduction of 12 % in the length of the ciphertext was obtained at a cost to the encrypting and decrypting complexities.
This thesis is structured as follows:

- Chapter 2 provides the necessary background on error-correcting codes, lattices, fundamental cryptographic concepts, including symmetric, asymmetric, and post-quantum cryptography, and an overview of lossy source coding principles relevant to quantization.
- In Chapter 3, we present a novel code-based analog of the quantized lattice-based symmetric cryptosystem from [MS23], detailing its construction, the application of quantization using codes, and an analysis of the resulting plaintext-to-ciphertext rates.
- Chapter 4 focuses on the HQC cryptosystem. It reviews the standard HQC scheme, its error model, and decryption failure rate considerations. We then introduce a quantized version of HQC, outline a search strategy for selecting optimal underlying codes for quantization, and present the results of this approach in terms of ciphertext size reduction.
- In Chapter 5, we summarize the findings of this research, discuss the implications of using quantization in code-based cryptography, and suggest potential directions for future work.

Through this investigation, we demonstrate that quantization can be valuable in reducing the ciphertext lengths often associated with code-based cryptography, thereby enhancing its practicality for future PQC applications.

2 Preliminaries

In this chapter, we will establish the mathematical basis for describing the cryptosystems considered in the later chapters. Firstly, notations used for the rest of the thesis are introduced in Section 2.1. As a background for the code-based cryptosystems, a brief recap of coding theory is presented in Section 2.2. In Section 2.3, select topics in lattices which form the basis for lattice-based cryptography are described. A high-level overview of symmetric, asymmetric, and post-quantum cryptography is discussed in Section 2.4. Finally, Section 2.5 discusses key results from lossy source coding theory, which is used to analyze quantization.

2.1 Notation

The Galois field of size q is represented by $\mathbb{F}_q$, the ring of integers of size q by $\mathbb{Z}_q$, and the polynomial ring of length n over $\mathbb{F}_2$ is $\mathcal{R}_n = \mathbb{F}_2[z]/(z^n - 1)$. We will use bold lowercase letters to represent vectors $\mathbf{x} = (x_0, \ldots, x_{n-1})$ and we say $\mathbf{x} \in \mathbb{F}_q^n$ when the elements $x_i \in \mathbb{F}_q$. Uppercase bold letters are used for matrices (for e.g.: $\mathbf{G}$) and similarly have elements from a field or a ring. Sampling a variable from a probability distribution χ is indicated as $V \leftarrow \chi$, and when the distribution is uniform over a set S, we simply write $V \leftarrow S$. We write $\mathbf{v} \leftarrow \chi^n$ when the elements of a random vector $\mathbf{v}$ are drawn independently and identically from a distribution χ. A binary random variable V sampled from a Bernoulli distribution with parameter p means that the probability $\Pr[V = 1] = p$. The binary entropy function $H_2(p) = -p \log p - (1-p) \log(1-p)$. All logarithmic functions are taken to the base 2 unless explicitly specified.

2.2 Codes

Error-correcting codes, often just referred to as codes, are used to detect and correct errors in digital data transmission and storage. Introduced by Claude Shannon and Richard Hamming in the late 1940s [Sha48], they laid the foundation for reliable communication over noisy channels. Today, codes are central to modern technologies—from satellite and wireless communication to storage systems and cryptography, including post-quantum

schemes that rely on the hardness of decoding problems. In this section, we will define some basic quantities and objects from coding theory, some important code families, quantization using codes, and finally, some hard problems with codes.

2.2.1 Basics

We begin by defining some basic metrics used in coding theory, namely the Hamming weight and distance.

Definition 1 (Hamming Weight)**.** *The Hamming weight of a finite length vector* $\mathbf{x} \in \mathbb{F}^n$ *is defined as the number of non-zero entries in* $\mathbf{x}$*.*

$$wt_H(\mathbf{x}) \triangleq |\{i \in \{0, \ldots, n-1\} | x_i \neq 0\}|$$

Definition 2 (Hamming Distance)**.** *The Hamming distance between two vectors* $\mathbf{x}, \mathbf{y} \in \mathbb{F}^n$ *is the Hamming weight of their difference.*

$$d_H(\mathbf{x}, \mathbf{y}) \triangleq wt_H(\mathbf{x} - \mathbf{y})$$

Definition 3 (Linear Block Codes)**.** *A linear block code* $\mathcal{C}$ *of length* n *and dimension* k *is a* k*-dimensional subspace of a vector space* $\mathbb{F}_q^n$*, often denoted* $[n, k, d]_q$ *to indicate the length, dimension, and minimum distance. The elements of* $\mathcal{C}$ *are called codewords, and the minimum distance* d *of* $\mathcal{C}$ *is the smallest weight among all the codewords.*

$$d = \min_{\mathbf{c} \in \mathcal{C}} wt_H(\mathbf{c})$$

The code rate R *quantifies the relative size of the message against that of the codeword.*

$$R = \frac{\log |\mathcal{C}|}{n} = \frac{k}{n}$$

In this work, we only consider linear block codes and refer to them as codes interchangeably.

Definition 4 (Generator and Parity Check Matrices)**.** *A matrix containing rows of basis vectors of* $\mathcal{C}$ *is called a generator matrix for* $\mathcal{C}$*. The code* $\mathcal{C}$ *can be defined using the generator matrix* $\mathbf{G} \in \mathbb{F}_q^{k \times n}$*.*

$$\mathcal{C} = \{\mathbf{x}\mathbf{G} | \mathbf{x} \in \mathbb{F}_q^k\}$$

Alternatively, $\mathcal{C}$ *can be defined using the parity-check matrix* $\mathbf{H} \in \mathbb{F}_q^{n-k \times n}$*.*

$$\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_q^n | \mathbf{c}\mathbf{H}^T = \mathbf{0}\}$$

In information theory, a channel describes the transformation of an input symbol x into an output symbol y based on some transition probability distribution $\Pr[y|x]$, which is the mathematical model of a real communication channel. One of the simplest channels is a Binary Symmetric Channel (BSC), which is defined for a binary input and output as

$$\Pr[y|x] = \begin{cases} 1-p & \text{if } y = x \\ p & \text{otherwise} \end{cases}$$

for some parameter $p \in [0, 1]$. Such a channel is represented as BSC(p), and one can view the transformation from x to y alternatively as adding some noise $e \leftarrow \text{Bernoulli}(p)$ such that $y = x + e \mod 2$.
For reliable communication over a channel, one needs to add redundancy to any transmission so that the receiver can recover the message accurately even in the event of noise. This can be achieved using codes where the code rate determines how much information or redundancy is contained in a codeword. In 1948, Shannon proved the channel coding theorem [Sha48], which states that for any channel, there exists a maximum code rate, called the channel capacity, below which the probability of making a decoding error at the receiver tends to 0 asymptotically. For a BSC(p), the channel capacity is

$$C_{BSC} = 1 - H_2(p) \tag{2.1}$$

2.2.2 Some Important Codes

There exist an abundant number of codes that have been developed since the inception of coding theory, of which we use only four families in this thesis.
The Reed-Muller (RM) and Reed-Solomon (RS) code families are used in the construction of the asymmetric cryptosystem HQC in Chapter 4.
Polar codes are attractive for quantization due to their low encoding and decoding complexities, as well as asymptotic guarantees.
A complete treatment of these codes would be out of the scope of this thesis. Instead, we refer the reader to the excellent book from Roth [Rot06] for RM and RS codes and the seminal paper from Arikan [Ari09] on polar codes.
The fourth family of codes are called quasi-cyclic codes and are only relevant here for the proof of hardness for some of the cryptosystems. Before we describe them, we must understand what cyclic codes are.

Definition 5 (Cyclic Codes)**.** *Let $\mathcal{C} \subset \mathbb{F}_q^n$ be a linear block code. We say that $\mathcal{C}$ is cyclic if and only if*

$$\forall \mathbf{c} \in \mathcal{C}, \quad \sigma(\mathbf{c}) \in \mathcal{C}$$

where for any

$$\mathbf{c} = (c_0, \ldots, c_{n-1}), \quad \sigma(\mathbf{c}) \triangleq (c_{n-1}, c_0, \ldots, c_{n-2})$$

Cyclic codes have circulant generator matrices.

$$\mathbf{G} = rot(\mathbf{g}) = \begin{bmatrix} \mathbf{g} \\ \sigma(\mathbf{g}) \\ \vdots \\ \sigma^{k-1}(\mathbf{g}) \end{bmatrix}$$

where $\mathbf{g}$ *is one of the basis vectors.*

Cyclic codes lend themselves to an alternative definition where the codewords can be represented by polynomials $g(x)$ from the ring $\mathcal{R}_n = \mathbb{F}_q[x]/(x^n - 1)$. A cyclic shift of $g(x)$ is $xg(x)$. And any codeword can be generated by the generator polynomial $g(x)$ as $u(x)g(x)$ where $u(x)$ is a degree k polynomial. The product of any two polynomials $a(x), b(x) \in \mathcal{R}_n$ can be represented as $\mathbf{a} \cdot rot(\mathbf{b})$ and vice versa. Here we have implicitly assumed the mapping between $\mathbf{a} = (a_0, \ldots, a_{n-1})$ and $a(x) = \sum_{i=0}^{n-1} a_i x^i$ in the natural order of the coefficients.

Definition 6 (Quasi-Cyclic Codes)**.** *A linear block code* $\mathcal{C}$ *is quasi-cyclic if*

$$\forall(\mathbf{c}_1, \mathbf{c}_2) \in \mathcal{C} \implies (\sigma(\mathbf{c}_1), \sigma(\mathbf{c}_2)) \in \mathcal{C}$$

The generator matrix of such a code has a double-circulant structure.

$$\mathbf{G} = [rot(\mathbf{g}_1), rot(\mathbf{g}_2)]$$

2.2.3 Code-based Quantization

Quantization, or more specifically vector quantization, is a technique from information theory that is used to compress long vectors into shorter representations. Later, we will extensively compress various vectors using quantization. In this subsection, we will shortly describe the notation used for quantization performed using codes. Section 2.5 gives a more detailed explanation of the theoretical limits of quantization. We can quantize any vector in $\mathbb{F}_q^n$ using a code $\mathcal{C}$ by mapping it to the nearest codeword with respect to the Hamming distance.

Definition 7 (Code-based Quantization)**.** *Given a code* $\mathcal{C} \subset \mathbb{F}_q^n$ *one can define the quantization function*

$$\lfloor \cdot \rceil_{\mathcal{C}} : \quad \mathbb{F}_q^n \mapsto \mathcal{C}$$

such that

$$\lfloor \mathbf{0} \rceil_{\mathcal{C}} = \mathbf{0}$$

and for any $\mathbf{c} \in \mathcal{C}$ *and* $\forall \mathbf{x} \in \mathbb{F}_q^n$

$$\lfloor \mathbf{x} + \mathbf{c} \rceil_{\mathcal{C}} = \lfloor \mathbf{x} \rceil_{\mathcal{C}} + \mathbf{c}$$

Correspondingly, the quantization error is defined as

$$\{\mathbf{x}\}_{\mathcal{C}} = \mathbf{x} - \lfloor \mathbf{x} \rceil_{\mathcal{C}} .$$

Any codeword can be uniquely represented by a k-length information vector, and since $k \leq n$ by the definition of a code, we can effectively compress the input vector. The lower the rate of the code is, the shorter the compressed representation will be, at the cost of a higher quantization error [Cov99].

2.2.4 Hard Problems on Codes

Problems in computer science can be classified into several categories based on the difficulty of solving them with a traditional computer. Problems like solving a linear system of equations are considered to be in the set of P, which contains all those that can be solved in a polynomial number of steps. NP problems can be verified in polynomial time. NP-hard problems are special because if one can solve an NP-hard problem in polynomial time, it means that every problem in NP can be solved in polynomial time. One of the millennium prize problems asks whether P and NP are the same, that is, whether an NP-hard problem can be solved in polynomial time. So far, this has turned out to be not true, and indeed, we only have exponential time solutions to NP-hard problems [Sip13]. The study of NP-hard problems is relevant to cryptography as they can be used to prove that an eavesdropper cannot decrypt the communication efficiently. Several NP-hard problems involve codes, and some of the relevant ones are introduced below.

Definition 8 (Syndrome Decoding Problem (SDP))**.** *Given* $\mathbf{s} \in \mathbb{F}_q^{n-k}$ *and* $\mathbf{H} \in \mathbb{F}_q^{n-k \times n}$*, find* $\mathbf{e} \in \mathbb{F}_q^n$ *such that* $\mathbf{e}\mathbf{H}^T = \mathbf{s}$ *and* $wt_H(\mathbf{e}) \leq t$ *for some integer* t*.*

Consider that a codeword $\mathbf{x}$ was sent over a channel where the noise $\mathbf{e}$ of weight at most t was added, and the received word $\mathbf{y} = \mathbf{x} + \mathbf{e} \mod 2$. We can find the syndrome $\mathbf{s} = \mathbf{y}\mathbf{H}^T = \mathbf{e}\mathbf{H}^T$ where we have used the definition of a parity-check matrix. Simply put, SDP asks to find a suitable error vector $\mathbf{e}$ for which we obtain the given syndrome $\mathbf{s}$. The following variant of SDP focuses only on quasi-cyclic codes.

Definition 9 (Quasi-Cyclic Syndrome Decoding Problem (QCSDP))**.** *Given* $\mathbf{s} \in \mathcal{R}_n$ *and* $\mathbf{h} \in \mathcal{R}_n$*, find* $\mathbf{e}_1, \mathbf{e}_2 \in \mathcal{R}_n$ *such that* $\mathbf{e}_1 + \mathbf{e}_2 \cdot \mathbf{h} = \mathbf{s}$ *and* $wt_H(\mathbf{e}_1) + wt_H(\mathbf{e}_2) \leq t$ *for some integer* t*.*

For some integer-valued functions t of n, the above problems are NP-hard [BMvT78]. The best-known attacks against these problems are derivatives of the Information Set Decoding algorithm from Prange [CTS16]. These problems are important because their hardness is used as the basis for several code-based cryptosystems, including HQC.
Another interesting problem is called Learning Parity with Noise (LPN), which is a variation of the regular SDP where now the error is not restricted by weight; instead it is sampled from a probability distribution.

Definition 10 (LPN)**.** *Given* $(\mathbf{G}, \mathbf{sG} + \mathbf{e})$ *find* $\mathbf{s}$*, where*

$$\mathbf{G} \leftarrow \mathbb{F}_q^{k \times n}$$

$$\mathbf{s} \leftarrow \mathbb{F}_q^k$$

$$\mathbf{e} \leftarrow \mathit{Bernoulli}(\tau)^n$$

for n *being a polynomial function of* k *and constant* τ*.*

Note that here $\mathbf{sG}$ is a codeword of the code defined by $\mathbf{G}$, and the elements of the error $\mathbf{e}$ are sampled independently and identically from the Bernoulli distribution. LPN will be relevant for the hardness and choice of parameters for the code-based symmetric cryptosystem presented in Chapter 3.

2.3 Lattices

Lattices are fascinating mathematical objects for constructing post-quantum cryptographic schemes. The finalist of the post-quantum cryptography competition from NIST is a key encapsulation mechanism based on lattices. As they are a source of inspiration for some of the ideas used in this work, let us discuss the basics of lattices, quantization, and hard problems on lattices.

2.3.1 Basics

Definition 11 (Lattices)**.** *A lattice is a discrete subgroup* $L = span_{\mathbb{Z}}(\mathbf{B}) \subset \mathbb{R}^n$ *for some basis* $\mathbf{B} \in \mathbb{R}^{n \times n}$*. The fundamental region of a lattice is the set* $\mathcal{F}(L) \subset \mathbb{R}^n$ *which satisfies*

$$|(\mathbf{x} + L) \cap \mathcal{F}(L)| = |\mathbf{x}| = 1 \quad \forall \mathbf{x} \in \mathbb{R}^n$$

.

Often, lattices are defined with respect to addition and multiplication modulo q for some large integer q. The advantage is that elements of such a q-ary lattice can be represented by a vector of integers coming from a bounded set, thus enabling them to be easily handled by computers. Here, we define a pair of functions that are used to map and de-map any vector $\mathbf{m}$ from $\mathbb{Z}_q^n$ to a lattice.

Definition 12 (Lattice Mapping)**.** *For a lattice $L \subset \mathbb{R}_q^n$ where q and n are integers, any vector $\mathbf{m} \in \mathbb{Z}_q^n$ can be "encoded" or mapped to a unique element of L.*

$$encode_L(\mathbf{m}) \triangleq \mathbf{Bm} \mod q$$

Here, $\mathbf{B}$ is some fixed basis of L. In the same note, we can define the de-mapping or "decoding" function for some vector $\mathbf{c} \in \mathbb{Z}_q^n$.

$$decode_L(\mathbf{c}) \triangleq \mathbf{B}^{-1} \lfloor \mathbf{c} \rceil_L \mod q$$

2.3.2 Lattice-based Quantization

Quantization of a vector from $\mathbb{R}^n$ can be performed using a lattice L by mapping it to the closest vector in L in terms of the Euclidean distance.

Definition 13 (Lattice-based Quantization)**.** *Given a lattice $L \subset \mathbb{R}^n$ one can define the quantization function*

$$\lfloor \cdot \rceil_L : \quad \mathbb{R}^n \mapsto L$$

such that

$$\lfloor \mathbf{0} \rceil_L = \mathbf{0}$$

and for any $\mathbf{v} \in L$ and $\forall \mathbf{x} \in \mathbb{R}^n$

$$\lfloor \mathbf{x} + \mathbf{v} \rceil_L = \lfloor \mathbf{x} \rceil_L + \mathbf{v} \quad \in L$$

Correspondingly, the quantization error is defined as

$$\{\mathbf{x}\}_L = \mathbf{x} - \lfloor \mathbf{x} \rceil_L \quad \in \mathcal{F}(L)$$

2.3.3 Hard Problems on Lattices

NP-hard problems involving lattices also exist. The Learning With Errors (LWE) problem is at least as hard as some worst-case hard lattice problems; hence, it is suitable to form

the basis of hardness for cryptographic schemes. In the seminal paper by Regev [Reg09], LWE was used to create lattice-based cryptosystems.

Definition 14 (LWE)**.** *Given* $(\mathbf{G}, \mathbf{sG} + \mathbf{e})$ *find* $\mathbf{s}$*, where*

$$\mathbf{G} \leftarrow \mathbb{Z}_q^{k \times n}$$

$$\mathbf{s} \leftarrow \mathbb{Z}_q^k$$

$$\mathbf{e} \leftarrow \chi^n$$

for n *being a polynomial function of* k

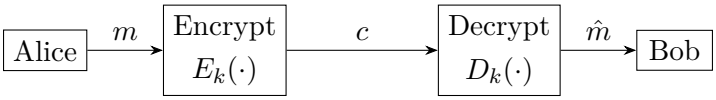
Here, χ^n is a distribution over $\mathbb{Z}_q^n$ which is concentrated around $\mathbf{0}$. In the code world, the analog of LWE is LPN, which is also conjectured to be an average-case hard problem [Pie12]. The integer rings are swapped for fields, and χ is instead a Bernoulli distribution of some small parameter $\tau \in [0, 1]$.

2.4 Cryptography

Cryptography is the study of techniques for secure communication in the presence of adversaries. Broadly, cryptographic schemes can be categorized as symmetric or asymmetric (also known as public-key cryptography). Post-quantum cryptography is currently an area of great interest due to the existence of quantum algorithms that can efficiently attack the current public-key algorithms given a sufficiently large quantum computer. Here, we provide a brief introduction to them, however a more complete treatment can be found in [KL07]. There are several notions of security for cryptographic schemes such as Indistinguishable under Chosen Plaintext Attack, Indistinguishable under Chosen Ciphertext Attack (IND-CCA) [KL07], and Pseudorandom under Chosen Plaintext Attack (RND-CPA) [MS23, Definition 9], and we refer the reader to the cited sources for more information.

2.4.1 Symmetric Cryptography

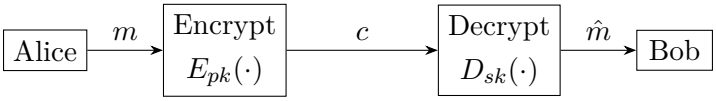
In symmetric cryptography, the same (secret) key k is used for both encryption and decryption. Consider two parties – Alice and Bob – where Alice wants to send a secret message to Bob. She encrypts the plaintext message m into a ciphertext c using the shared key k. Bob uses the same key to decrypt the ciphertext back into the plaintext.



Examples include the Advanced Encryption Standard (AES) and ChaCha20 [Sta23]. Symmetric cryptosystems are efficient and well-suited for bulk data encryption. The challenge is to distribute the secret key to both parties securely. This is achieved by key encapsulation mechanisms (KEMs) based on asymmetric or public-key cryptography.

2.4.2 Asymmetric Cryptography

Asymmetric cryptography uses a key pair consisting of a public key pk for encryption and a secret key sk for decryption. This eliminates the need for prior shared secrets and enables functionalities such as digital signatures and KEMs for symmetric cryptosystems.



Well-known asymmetric schemes include RSA, ElGamal, and ECC. These schemes use the conjectured hardness of the integer factorization problem or the discrete logarithm problem [KL07, Definition 8.62] to ensure security. While no algorithm is known to be able to attack these schemes in polynomial time on a traditional computer, Shor's algorithm [Sho99] uses a quantum computer to attack them in polynomial time. Due to their resistance to quantum attacks, post-quantum cryptosystems are of great interest, especially code-based (e.g., McEliece, HQC), lattice-based (e.g., Kyber), and multivariate cryptosystems.

2.4.3 Post-Quantum Cryptography

The advent of quantum computing poses significant threats to classical cryptographic systems hence robust alternatives are needed. Lattice-based cryptography relies on the presumed hardness of mathematical problems over lattices such as LWE. Well-known lattice-based schemes include KEMs like CRYSTALS-Kyber [ABD+20], Module-Lattice-Based Key Encapsulation Mechanism (ML-KEM), and the digital signature scheme Dilithium. Code-based cryptography leverages the computational difficulty of decoding random linear codes, specifically the SDP. The first such scheme, the McEliece cryptosystem [McE78], remains unbroken, despite recent improvements in attacks against it [Ran25].

Several other code-based cryptographic schemes were later proposed, such as the Niederreiter [Nie86] cryptosystem and the HQC cryptosystem, which is analogous to CRYSTALS-Kyber. In contrast, symmetric schemes such as AES remain relatively secure but require larger key sizes due to Grover's quantum algorithm, which offers only a quadratic speedup. One of the disadvantages of post-quantum cryptosystems is that the size of the ciphertexts are much longer than traditional algorithms. The relative sizes of the plaintext and ciphertext can be quantified using the "plaintext-to-ciphertext" rate. This quantity is helpful to compare the efficiency of various cryptographic schemes.

Definition 15 (Plaintext-to-Ciphertext Rate)**.** *For any cryptographic algorithm that accepts a plaintext input from the set M and outputs a ciphertext from the set C, the plaintext to ciphertext rate is defined as*

$$\alpha = \frac{\log |M|}{\log |C|} \tag{2.2}$$

It compares the number of bits needed to represent the plaintext to the number of bits needed to represent the ciphertext.

2.5 Lossy Source Coding

In information theory, source coding is the study of the fundamental limits of compressing data. Shannon proved in 1948 [Sha48] the source coding theorem, which states that a sequence of n bits can be theoretically compressed to a length of ρn bits without loss in information, where $\rho = H_2(p)$. Here, $H_2(p) \triangleq -p \log_2 p - (1-p)\log_2(1-p) \in [0,1]$ is the entropy function and $p = \Pr[X = 1]$ for the random variable X representing the binary memoryless source. If one is willing to bear some deterioration of the sequence, the compression can be improved further. This is studied in lossy source coding. This deterioration is quantified by a *distortion* measure $d(\cdot,\cdot)$. Generally, the normalized Hamming distance is used for bit sequences.

$$d(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{n} \sum_{i=0}^{n-1} d_H(x_i, \hat{x}_i)$$

where $\mathbf{x}, \hat{\mathbf{x}} \in \mathbb{F}_2^n$. The goal of lossy source coding is to minimize the compression factor ρ such that the output sequence has an average distortion at most some constant D.

$$R(D) = \min_{P_{X|\hat{X}}} R \quad \text{s.t.} \quad \mathbb{E}[d(X, \hat{X})] \leq D$$

The function $R(D)$ is called the rate-distortion bound, and it gives a lower bound on the best compression one can achieve at a specified average distortion. For a uniform binary source X, the rate-distortion bound is

$$R(D) = 1 - H_2(D) \tag{2.3}$$

$R(D)$ represents an asymptotic theoretical limit for compression, and a detailed derivation can be found in [Cov99, Chapter 10]. Often, in practice, this limit is not reached for finite block lengths and efficient source encoding algorithms. For a finite block-length n, the theoretical rate-distortion bound was shown in [KV12] to be approximately

$$R_n(D) = 1 - H_2(D) + \frac{\log n}{n} + O\left(\frac{1}{n}\right) \tag{2.4}$$

There are several practical source encoder designs, but we will focus on the one presented in [KU10], which uses the successive cancellation decoder of a polar code for compression. While it attains the rate-distortion bound asymptotically with quasi-linear encoding and decoding complexity, for finite block-lengths it deviates from the limit. The achieved average distortion rate is upper bounded by $D + O(2^{-N^\beta})$ where $\beta \in (0, \frac{1}{2})$ is a parameter which can be chosen and $N = 2^n$ is the block length of the polar code for some natural number n. The complexities of encoding (which uses a successive cancellation polar decoder) and decoding (which uses the polarization transform) are both $O(n \log n)$ [Ari09].

3 Symmetric Cryptosystems

The goal of this thesis is to reduce the ciphertext sizes of various code-based cryptosystems through quantization. As a first step, we consider symmetric cryptosystems as they are simpler to analyze and exhibit good properties with quantization. We take inspiration from the quantized lattice-based symmetric cryptosystem proposed by [MS23] and apply the same framework to codes. In Section 3.1, we summarize the quantized symmetric cryptosystem in lattices by [MS23]. Then, we show how a code-based analog of this cryptosystem can be constructed in Section 3.2. The performance of this system is discussed in Section 3.3.

3.1 Quantized Lattice-based Symmetric Cryptosystem

In [MS23], a simple symmetric cryptosystem is used to demonstrate the effect of quantization. Then, they apply the same techniques to lattice-based asymmetric cryptosystems and analyze the ciphertext length. They also show how the parameters of their cryptosystem can be chosen to optimize the plaintext-to-ciphertext rate asymptotically.

3.1.1 A Simple Lattice-based Symmetric Cryptosystem

First, let us consider their system without quantization as shown in Table 3.1. A lattice E is used to encode the message.

Table 3.1: Lattice-based symmetric cryptosystem by [MS23].

$KeyGen(1^k)$	$Encrypt_{\mathbf{s}}(\mathbf{m})$	$Decrypt_{\mathbf{s}}(\mathbf{G}, \mathbf{b})$
$\mathbf{s} \leftarrow \mathbb{Z}_q^k$	$\mathbf{G} \leftarrow \mathbb{Z}_q^{k \times n}$	return $decode_E(\mathbf{b} - \mathbf{sG})$
return $\mathbf{s}$	$\mathbf{e} \leftarrow \chi^n$	
	$\mathbf{b} = \mathbf{sG} + \mathbf{e} + encode_E(\mathbf{m})$	
	return $(\mathbf{G}, \mathbf{b})$	

Here, the error $\mathbf{e}$ comes from a probability distribution χ^n which is concentrated around $\mathbf{0}$. Usually it is chosen to be the (discrete) Gaussian distribution. An example of this scheme is visualized in Figure 3.1 for the two-dimensional case.

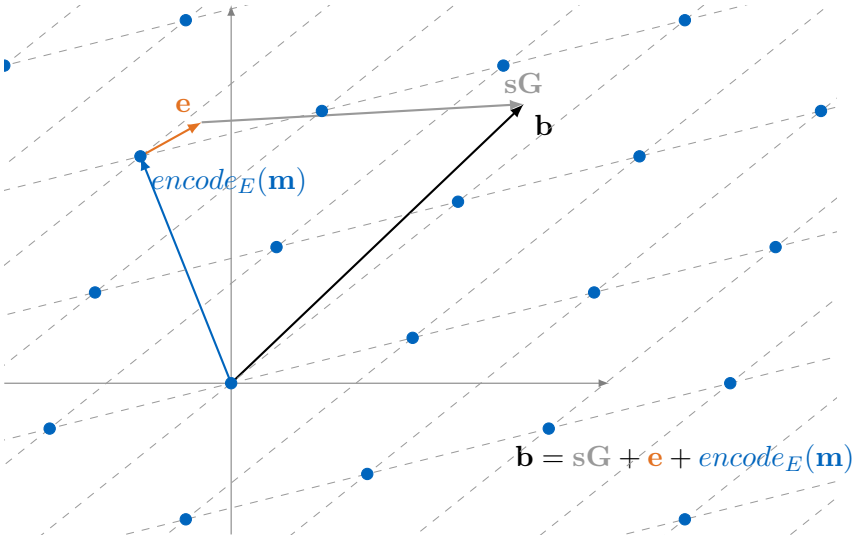


Figure 3.1: An example of a two-dimensional lattice in $\mathbb{R}^2$ is shown here in blue dots. The elements of the cryptosystem are shown as vectors.

3.1.2 Applying Quantization

Now, the ciphertext length of the scheme presented in Table 3.1 can be improved with quantization using a lattice $Q \subset \mathbb{R}^n$. To quantize a vector $\mathbf{c}$, one must solve the Closest-Vector Problem for Q and store the corresponding lattice point. The quantized scheme is presented in Table 3.2. Most of the steps remain the same as before, except that an additional quantization step is added, which is visualized in Figure 3.2.

Table 3.2: Lattice-based symmetric cryptosystem with quantization by [MS23].

$KeyGen(1^k)$	$Encrypt_{\mathbf{s}}(\mathbf{m})$	$Decrypt_{\mathbf{s}}(\mathbf{G}, \lfloor \mathbf{b} \rceil_Q)$
$\mathbf{s} \leftarrow \mathbb{Z}_q^k$	$\mathbf{G} \leftarrow \mathbb{Z}_q^{k \times n}$	return $decode_E(\lfloor \mathbf{b} \rceil_Q - \mathbf{sG})$
return $\mathbf{s}$	$\mathbf{e} \leftarrow \chi^n$	
	$\mathbf{b} = \mathbf{sG} + \mathbf{e} + encode_E(\mathbf{m})$	
	return $(\mathbf{G}, \lfloor \mathbf{b} \rceil_Q)$	

Let us look closer at the decryption step from Table 3.2.

$$\begin{aligned} decode_E(\lfloor \mathbf{b} \rceil_Q - \mathbf{sG}) &= decode_E(\mathbf{b} - \{\mathbf{b}\}_Q - \mathbf{sG}) \\ &= \mathbf{m} + decode_E(\mathbf{e} - \{\mathbf{b}\}_Q) \end{aligned}$$

For the decryption to be correct, the error term $\mathbf{e} - \{\mathbf{b}\}_Q$ must decode to $\mathbf{0}$ in E. That

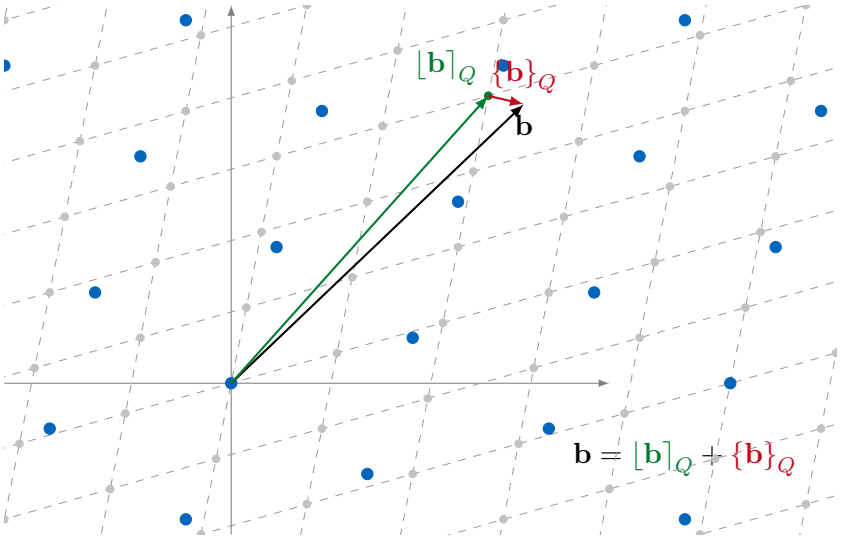


Figure 3.2: The quantization is performed with another two-dimensional lattice Q represented with grey dots. The effect of quantization is to map every vector $\mathbf{b}$ to the nearest lattice point from Q represented by $\lfloor \mathbf{b} \rceil_Q$. This also introduces an error, which is shown here in red $\{\mathbf{b}\}_Q$.

is to say that the error term must belong to the fundamental region $\mathcal{F}(E)$.

We say that the scheme is δ-correct when the probability that the decryption is incorrect is at most $\delta \geq 0$. Thus, a 0-correct scheme always performs correct decryption, which is ideal. This δ can be controlled by choosing the lattices E and Q appropriately, given a fixed error distribution χ. The size of the fundamental region of E determines the number of vectors that are mapped to $\mathbf{0}$ with the $decode_E$ function. Note that the quantization error $\{\mathbf{b}\}_Q$ lies within the fundamental region $\mathcal{F}(Q)$ while the error $\mathbf{e}$ is concentrated around $\mathbf{0}$. When χ is a bounded set, we can choose E with a fundamental region $\mathcal{F}(E)$ that envelopes the convex sum of the sets $\mathcal{F}(Q)$ and the support of χ^n to achieve $\delta = 0$.

In [MS23], it is shown that for such a "bounded noise" model, the plaintext-to-ciphertext rate $\alpha = 1 - o(1)$ can be achieved by quantization. This means that with even a little bit of quantization, one can achieve ciphertext sizes, which are the same as plaintext sizes asymptotically. The size of $\mathbf{G}$ is not included in the computation of α as its overhead can be mitigated through standard techniques. In practice, the cost of transmitting $\mathbf{G}$ is amortized over multiple encryptions, and a full matrix is rarely sent; instead, a short seed is transmitted, which can be deterministically expanded to reconstruct $\mathbf{G}$. Usually, the seed is chosen to be of length λ, where λ is the required security level of the scheme.

3.1.3 Dithering

The two components of the error term $\mathbf{e}$ and $\lfloor \mathbf{b} \rceil_Q$ are dependent on each other, and moreover, it is hard to describe the distribution of $\lfloor \mathbf{b} \rceil_Q$. This makes the analysis of the scheme difficult. Inspired by a concept in signal processing, dithering is introduced in [MS23] in order to separate the two error components so that they can be treated individually. A dither is a uniformly random vector that is added to $\mathbf{b}$ during encryption. The complete scheme with quantization and dithering is presented in Table 3.3.

Table 3.3: Lattice-based symmetric cryptosystem with quantization and dithering by [MS23].

$KeyGen(1^k)$	$Encrypt_{\mathbf{s}}(\mathbf{m})$	$Decrypt_{\mathbf{s}}(\mathbf{G}, \lfloor \mathbf{b}-\mathbf{v} \rceil_Q, \mathbf{v})$
$\mathbf{s} \leftarrow \mathbb{Z}_q^k$	$\mathbf{G} \leftarrow \mathbb{Z}_q^{k \times n}$	return $decode_E(\lfloor \mathbf{b}-\mathbf{v} \rceil_Q + \mathbf{v} - \mathbf{sG})$
return $\mathbf{s}$	$\mathbf{e} \leftarrow \chi^n$	
	$\mathbf{b} = \mathbf{sG} + \mathbf{e} + encode_E(\mathbf{m})$	
	$\mathbf{v} \leftarrow \mathcal{F}(Q)$	
	return $(\mathbf{G}, \lfloor \mathbf{b}-\mathbf{v} \rceil_Q, \mathbf{v})$	

Now, for decoding correctness $\mathbf{e} - \{\mathbf{b}-\mathbf{v}\}_Q$ must decode to $\mathbf{0}$. Since $\mathbf{v}$ is uniformly random over $\mathcal{F}(Q)$, it ensures that

1. $\lfloor \mathbf{b}-\mathbf{v} \rceil_Q$ is uniform over $\mathcal{F}(Q)$, and
2. $\mathbf{e}$ is independent from $\lfloor \mathbf{b}-\mathbf{v} \rceil_Q$.

The dithering vector $\mathbf{v}$ can also be generated using a short seed, introducing only a relatively small overhead, which vanishes asymptotically as the length of the ciphertext grows. Thus, it is also not included in the calculation of the plaintext-to-ciphertext rate. The security of this scheme is proven using the hardness assumption of the LWE problem. In [MS23, Theorem 3], the dithered version of the scheme is proved to be RND-CPA secure. They also show that if $\mathbf{e}$ is restricted to $\mathbb{Z}_q^n$ instead of $\mathbb{R}_q^n$, care must be taken to ensure that only one of E and Q can be a lattice in $\mathbb{Z}_q^n$, that is, they should not share a common structure.

3.2 Quantized Code-based Symmetric Cryptosystem

In this section, we present our code-based symmetric cryptosystem in the style of the lattice-based scheme presented above. As seen in Table 3.4, the framework remains essentially the same, except for a few key differences, namely —

- we work over the binary field ($\mathbb{F}_2$) instead of the integer ring,
- the small error $\mathbf{e}$ is sampled from a set of *i.i.d* Bernoulli random variables, and
- E and Q are binary linear block codes.

Table 3.4: The novel code-based symmetric cryptosystem with quantization.

$KeyGen(1^k)$	$Encrypt_{\mathbf{s}}(m)$	$Decrypt_{\mathbf{s}}(\mathbf{G}, \lfloor \mathbf{b} \rceil_Q)$
$\mathbf{s} \leftarrow \mathbb{F}_2^k$	$\mathbf{G} \leftarrow \mathbb{F}_2^{k \times n}$	return $decode_E(\lfloor \mathbf{b} \rceil_Q - \mathbf{sG})$
return $\mathbf{s}$	$\mathbf{e} \leftarrow \text{Bernoulli}(\tau)^n$	
	$\mathbf{b} = \mathbf{sG} + \mathbf{e} + encode_E(\mathbf{m})$	
	return $(\mathbf{G}, \lfloor \mathbf{b} \rceil_Q)$	

Quantization in codes works similarly to lattices, and here we choose a polar code to perform the quantization due to its asymptotic guarantees and low encoding and decoding complexities [KU10]. The ciphertext $\mathbf{b}$ is fed as an input to a decoder of a polar code designed for a BSC with parameter p. This ensures that the average distortion of the quantizer is p (refer to Chapter 2, Section 2.5). The output of the decoder $\lfloor \mathbf{x} \rceil_Q$ is the nearest codeword in Q to $\mathbf{x}$. From this codeword, one stores only the information bits, resulting in a compression factor ρ, which is equal to the code rate $R_Q = \frac{k_Q}{n}$. A lower code rate leads to shorter ciphertexts, but it also increases the distortion. From the rate-distortion bound (Equation (2.3)), we have a lower bound on the compression factor achievable for a desired average distortion p.

$$\rho \geq 1 - H_2(p)$$

Polar codes have been shown to achieve the rate-distortion bound for a distortion value p asymptotically [KU10] when constructed for the channel BSC(p). In this paper, they also show that the quantization error can be approximated with a random vector of *i.i.d.* Bernoulli bits with the parameter p (Lemma 9). Concretely, the statistical distance between the distribution of the quantization error and Bernoulli$(p)^n$ is upper bounded by $O(2^{-N^\beta})$ for some $\beta \in (0, \frac{1}{2})$. We can thus approximate the quantization error as a BSC of parameter p for our analysis.

3.2.1 Correctness of Decryption

For a cryptosystem, it is important that decryption is correct with *overwhelming* probability for IND-CCA security [KL07]. In the decryption step, we use a decoder for the code

E, which must correct a noisy message. The noise is composed of the intentional noise $\mathbf{e}$ and the quantization error $\{\mathbf{b}\}_Q$. Combining the fact that we chose $\mathbf{e}$ from Bernoulli$(\tau)^n$ with the result from [KU10], we can form a representation for the channel that is "seen" by E.

$$\begin{aligned} decode_E(\lfloor \mathbf{b} \rceil_Q - \mathbf{sG}) &= decode_E(\mathbf{b} - \{\mathbf{b}\}_Q - \mathbf{sG}) \\ &= \mathbf{m} + decode_E(\mathbf{e} - \{\mathbf{b}\}_Q) \end{aligned}$$

As in the lattice case, the two components of the error seen by E are not independent of each other. We again use dithering [MS23] to add a uniform noise vector $\mathbf{v} \leftarrow \mathcal{F}(Q)$ which removes the dependence. Here, we borrow the concept of a fundamental region from lattices and apply it to codes as well. One way to define $\mathcal{F}(Q)$ is to fix a particular decoder for Q (say a successive cancellation decoder) and assign to $\mathcal{F}(Q)$ all those vectors $\mathbf{e} \in \mathbb{F}_2^n$ which are decoded to $\mathbf{0}$ under this decoder.

Definition 16 (Fundamental Region of a Code)**.** *Given a code $\mathcal{C} \subset \mathbb{F}_2^n$ with a decoder $decode_{\mathcal{C}}(\cdot)$,*

$$\mathcal{F}(\mathcal{C}) \triangleq \{\mathbf{e} \in \mathbb{F}_2^n \,|\, decode_{\mathcal{C}}(\mathbf{e}) = \mathbf{0}\}$$

The properties of this fundamental region are the same as in the case of lattices, and importantly, it lets us perform the dithering in the code-bases system as well. The sampled dither needs to be sent to the receiver along with the ciphertext for decryption to be possible, which adds an overhead on the ciphertext length, but we can mitigate this overhead by using a seed to generate the dither.

Table 3.5: The novel code-based symmetric cryptosystem with dithering. The dithering vector $\mathbf{v}$ removes dependence between $\mathbf{e}$ and the quantization error $\{\mathbf{b}\}_Q$

$KeyGen(1^k)$	$Encrypt_{\mathbf{s}}(m)$	$Decrypt_{\mathbf{s}}(\mathbf{G}, \lfloor \mathbf{b} - \mathbf{v} \rceil_Q, \mathbf{v})$
$\mathbf{s} \leftarrow \mathbb{F}_2^k$	$\mathbf{G} \leftarrow \mathbb{F}_2^{k \times n}$	return
return $\mathbf{s}$	$\mathbf{e} \leftarrow$ Bernoulli$(\tau)^n$	$decode_E(\lfloor \mathbf{b} - \mathbf{v} \rceil_Q + \mathbf{v} - \mathbf{sG})$
	$\mathbf{b} = \mathbf{sG} + \mathbf{e} + encode_E(\mathbf{m})$	
	$\mathbf{v} \leftarrow \mathcal{F}(Q)$	
	return $(\mathbf{G}, \lfloor \mathbf{b} - \mathbf{v} \rceil_Q, \mathbf{v})$	

Due to dithering, we can view the channel seen by E as a concatenation of two independent BSCs (Figure 3.3). The overall channel is again a BSC with a parameter $\tau_e = \tau(1-p) + (1-\tau)p$ [Cov99]. Thus, we need to choose a capacity-achieving code E

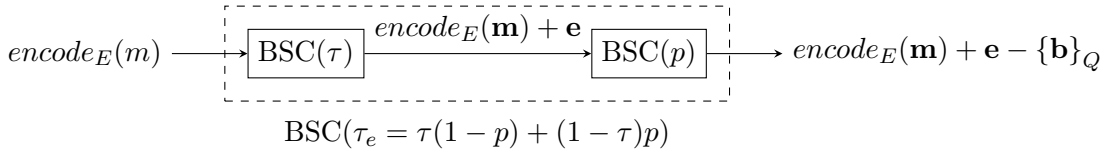


Figure 3.3: The effective channel "seen" by the code E at the decryption stage.

with a rate $R_E = 1 - H_2(\tau_e)$ for the best possible plaintext-to-ciphertext rate α.

3.3 Results

In Section 3.2, we presented a novel quantized code-based symmetric cryptosystem. We saw a model of the error that needs to be corrected by the message encoding code E and an approximation of the quantization error introduced by the quantization code Q. Now, let us analyze the performance of this cryptosystem by comparing the quantized and non-quantized versions. The plaintext-to-ciphertext rate of this code-based cryptosystem (with and without quantization) is

$$\alpha = \frac{R_E}{\rho}. \tag{3.1}$$

Without quantization, the compression factor is $\rho = 1$, and correspondingly, the average distortion is $p = 0$. This is the same as choosing a trivial code $[n, n, 1]_2$ for Q; hence there is no quantization error term. Code E thus needs to only correct the intentional error, that is, $\tau_e = \tau$. We can use any capacity-achieving code E over the BSC(τ) for this task. Plugging it into Equation (3.1), we get the plaintext-to-ciphertext rate for the non-quantized version of the scheme.

$$\alpha' = 1 - H_2(\tau)$$

With even a little bit of quantization, the compression factor ρ is smaller than 1, and this increases the plaintext-to-ciphertext rate. However, now the channel seen by E is a BSC of larger parameter τ_e and hence has a lower capacity $1 - H_2(\tau_e)$. We observe that for all pairs (τ, p), the plaintext-to-ciphertext rate with quantization α is at least as large as α' asymptotically.

$$\alpha = \frac{1 - H_2(\tau_e)}{1 - H_2(p)} \geq \alpha'$$

Figure 3.4 plots the plaintext-to-ciphertext rates for the system with and without quantization for $\tau = 0.1$. We clearly see that quantization leads to better rates even for small levels of added quantization error p.

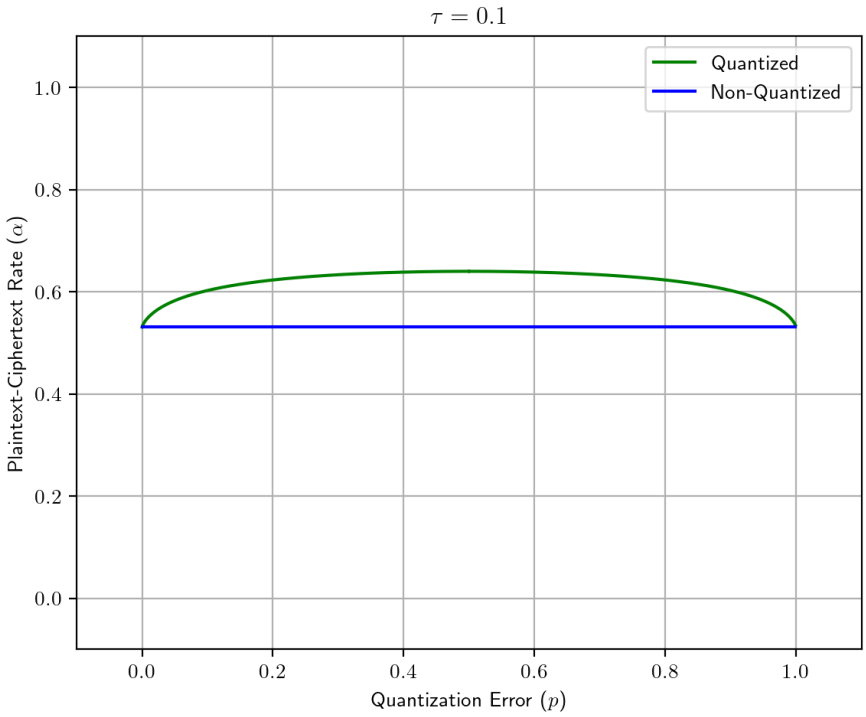


Figure 3.4: Asymptotic plaintext-to-ciphertext rate of the quantized (green) and non-quantized (blue) code-based symmetric cryptosystem.

So far, we have only considered the asymptotic behavior, but in reality, neither the quantizer Q nor the encoding code E would achieve capacity in the finite length regime. For finite block lengths, the quantizer will only achieve a sub-optimal compression factor $\rho = 1 - H_2(p) + \delta_\rho$. Moreover, the code E may only achieve a rate less than the channel capacity, that is to say that the rate $R_E = 1 - H_2(\tau_e) - \delta_{R_E}$. Together, they reduce the intervals of p for which a rate gain is observed. These regions can be observed in Figure 3.5. Here, $\alpha - \alpha'$ is plotted in color against τ and p. As δ_ρ is increased, the colored regions – which indicate the pairs (τ, p) for which the ciphertext length is shorter with quantization – shrink. Hence, it is important for the quantization code to be very close to the rate-distortion bound Equation (2.3).

The parameters of the scheme need to be chosen such that the LPN problem is hard. We choose n as a polynomial in k and fix τ as a constant, thus operating in the so-called constant-noise regime of the LPN assumption [YZW+17]. Fixing k has the benefit that the secret key size stays constant even if we choose to vary n.

Our scheme in Table 3.5 is correct when the decoding operation in E is correct. As we have observed in Figure 3.5, a capacity-achieving code E performs better in terms of plaintext-to-ciphertext rate. If we use a maximum-likelihood decoder for E, then we are sure to achieve this capacity (asymptotically) while ensuring that the probability of a decoding error tends to 0. Thus, asymptotically, this scheme is 0-correct. Practically, we can neither achieve capacity at finite lengths nor perform maximum-likelihood decoding. Hence, the performance of a practical version of our quantized code-based symmetric cryptosystem will be less than the optimal behavior shown here. One needs to make careful choices for the codes E and Q so that the desired correctness is obtained. The correctness of the scheme depends on E and its decoder, as well as the error introduced by the quantization code Q. An exact analysis of the correctness would require computing block error rates for specific choices of E and the decoder. Further, one would also need to simulate the actual quantization error rather than using the BSC approximation. Although not considered in this work, such an analysis may offer interesting insights and is left for future work.

Armed with an understanding of the behavior of quantization in this simple setting, we now explore the effects on a more complex and practical asymmetric cryptosystem, namely the HQC cryptosystem.

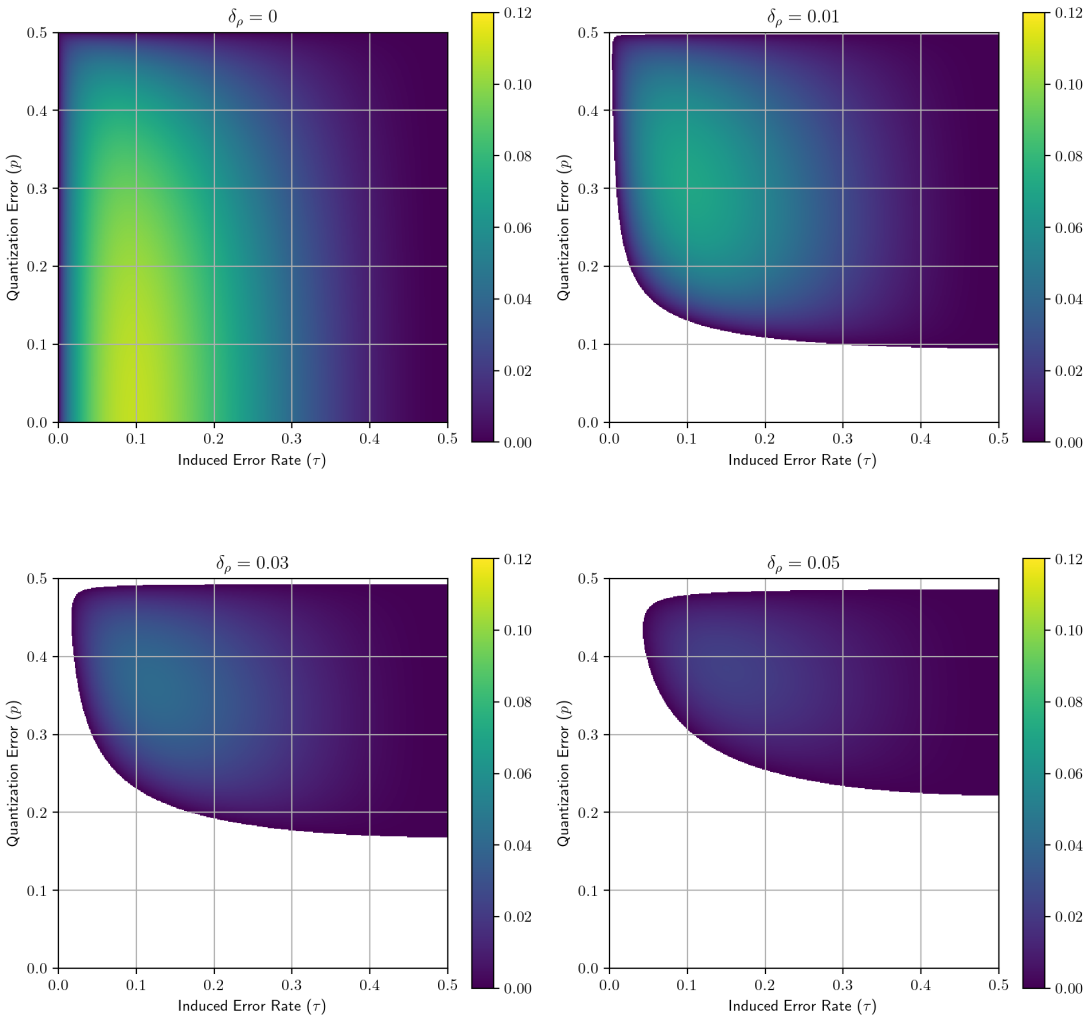


Figure 3.5: Difference in plaintext-to-ciphertext rate of the quantized and non-quantized code-based symmetric cryptosystem with compression factor $\rho = 1 - H_2(p) + \delta_\rho$ for various δ_ρ. As the ρ moves away from the rate-distortion limit, the region of (τ, p) pairs that produce $\alpha > \alpha'$ shrinks.

4 Asymmetric Cryptosystems

The primary reason for looking at post-quantum cryptography is the resistance to quantum attacks. While quantum algorithms reduce the complexity of an attack against classical symmetric cryptosystems like the AES, the quantum attacks against asymmetric cryptosystems such as RSA or ECC are even more powerful. Grover's algorithm [Gro96] is a quantum search algorithm that can find a target element in a list of n unsorted elements in $O(\sqrt{n})$ time instead of the classical $O(n)$. When applied to AES, this has the effect of halving the security level; for example, AES-256 would have only 128 bits of quantum security as opposed to the classical 256. Whereas for RSA, the complexity goes from exponential to polynomial time using Shor's algorithm [Sho99]. In this chapter, we will present an improvement to the current HQC cryptosystem [AMAB$^+$25]. Firstly, the scheme is summarized in Section 4.1, along with the construction of an appropriate error-correcting code for HQC. In Section 4.2, the quantized version of this scheme is presented, and strategies to search for optimal code constructions are discussed. Finally, the results obtained are showcased in Section 4.3.

4.1 HQC Cryptosystem

First put forth in 2018, the Hamming Quasi-Cyclic (HQC) cryptosystem is a code-based asymmetric cryptosystem (or a KEM) which is special because unlike the McEliece Cryptosystem or Niederreiter Cryptosystem, the underlying code is public. The security of HQC is due to the hardness assumption of the Quasi-Cyclic Syndrome Decoding Problem (QCSDP) problem [AMBD$^+$18]. In 2025, NIST selected HQC as a backup to the main general encryption algorithm called ML-KEM. We will briefly describe the scheme in Subsection 4.1.1, then we talk about the error model used and the correctness of decryption in Subsection 4.1.2, and finally, the Decryption Failure Rate is analyzed in Subsection 4.1.3.

4.1.1 The Scheme

An overview of the scheme is presented in Table 4.1.

Table 4.1: HQC Cryptosystem

$KeyGen(1^k)$	$Encrypt_{pk}(\mathbf{m})$	$Decrypt_{sk}(\mathbf{a}, \mathbf{c})$
sk: $\mathbf{s} \leftarrow \mathcal{A}_n(w_u)$	$\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3 \leftarrow \mathcal{A}_n(w_r)$	return
$\mathbf{g} \leftarrow \mathcal{R}_n$	$\mathbf{a} = \mathbf{r}_1 + \mathbf{g} \cdot \mathbf{r}_2$	$decode_{\mathcal{C}}(\mathbf{c} - \mathbf{a} \cdot \mathbf{s})$
$\mathbf{e} \leftarrow \mathcal{A}_n(w_u)$	$\mathbf{c} = \mathbf{b} \cdot \mathbf{r}_2 + \mathbf{r}_3 + encode_{\mathcal{C}}(\mathbf{m})$	
$\mathbf{b} = \mathbf{g} \cdot \mathbf{s} + \mathbf{e}$	return $(\mathbf{a}, \mathbf{c})$	
pk: $(\mathbf{g}, \mathbf{b})$		
return pk		

Here, we define

$$\mathcal{A}_n(w) \triangleq \{\mathbf{x} \in \mathcal{R}_n | wt_H(\mathbf{x}) = w\}.$$

The $encode_{\mathcal{C}}(\cdot)$ function maps the message to a codeword of $\mathcal{C}$. In the HQC specification, this is chosen to be a concatenated RM and RS code. This RMRS construction consists of a first-order RM code $[n_{RM}, k_{RM}, d_{RM}]_2$ and an RS code $[n_{RS}, k_{RS}, d_{RS}]_{2^{k_{RM}}}$. For a message length of k, the RM and RS codes are chosen such that $k = k_{RM}k_{RS}$. The resulting codeword is of length $n_{RM}n_{RS}$. The basic premise of HQC is that for the public code $\mathcal{C}$, the errors introduced in the encryption step are of weight higher than what can be corrected by the decoder chosen for $\mathcal{C}$. Only if the secret key $\mathbf{s}$ is known the weight of the error can be reduced enough that the decoder is able to correct it. This scheme does not deterministically guarantee that the weight of the error is reduced to a value low enough rather an analysis is made to find and ensure the Decryption Failure Rate (DFR) is smaller than $2^{-\lambda}$ where λ is the desired security level [AMAD+24].

4.1.2 Correctness and Error Model

For the correctness of decryption of this scheme, it is necessary that the error introduced $\tilde{\mathbf{e}} = \mathbf{s} \cdot \mathbf{r}_1 + \mathbf{e} \cdot \mathbf{r}_2 + \mathbf{r}_3$ must decode to $\mathbf{0}$ under the decoder for $\mathcal{C}$.

$$\begin{aligned} decode_{\mathcal{C}}(\mathbf{c} - (\mathbf{r}_1 + \mathbf{g} \cdot \mathbf{r}_2) \cdot \mathbf{s}) &= \mathbf{m} + decode_{\mathcal{C}}(\mathbf{g} \cdot \mathbf{s} \cdot \mathbf{r}_2 + \mathbf{e} \cdot \mathbf{r}_2 + \mathbf{r}_3 - \mathbf{r}_1 \cdot \mathbf{s} - \mathbf{g} \cdot \mathbf{r}_2 \cdot \mathbf{s}) \\ &= \mathbf{m} + decode_{\mathcal{C}}(\mathbf{e} \cdot \mathbf{r}_2 + \mathbf{r}_3 - \mathbf{r}_1 \cdot \mathbf{s}) \\ &= \mathbf{m} + decode_{\mathcal{C}}(\tilde{\mathbf{e}}) \end{aligned}$$

Above, we have used the fact that in the ring $\mathcal{R}_n = \mathbb{F}_2[z]/(z^n - 1)$ subtraction and addition are the same. In HQC, correctness cannot be guaranteed for all cases; that is,

with some probability (known as the Decryption Failure Rate (DFR)), the decryption will fail. An exact calculation of the DFR is difficult hence a simplification is made to the error model. The error $\tilde{\mathbf{e}}$ is approximated by a random vector of *i.i.d.* Bernoulli random variables with some parameter p^*. In reality, there is dependence between the bits of $\tilde{\mathbf{e}}$, but it is ignored for the sake of simplicity. Let us consider this assumption and DFR analysis in more detail.

Understanding the error $\tilde{\mathbf{e}} = \mathbf{s} \cdot \mathbf{r}_1 + \mathbf{e} \cdot \mathbf{r}_2 + \mathbf{r}_3$ is important for choosing an appropriate code $\mathcal{C}$. Let us break this down into three terms. Firstly, for the term $\mathbf{z} = \mathbf{s} \cdot \mathbf{r}_1$, the probability distribution of a single bit is

$$\tilde{p} \triangleq \Pr[z_k = 1] = \frac{1}{\binom{n}{w_u}\binom{n}{w_r}} \sum_{\substack{l=1 \\ l \text{ odd}}}^{\min(w_u, w_r)} \binom{n}{l}\binom{n-l}{w_u-l}\binom{n-w_u}{w_r-l}, \tag{4.1}$$

as derived in [AMAB$^+$25, Propositon 2.4.1].

Here, one can make the assumption that the bits of $\mathbf{z}$ are not dependent on each other, hence making it a random vector drawn from a Bernoulli$(\tilde{p})^n$ distribution. We know that the sum of two independent Bernoulli random variables is, again, a Bernoulli random variable. Combining this fact with the assumed distribution of the error terms, a formula for p^*, which is the Bernoulli parameter for a single bit $\tilde{e}_k$ of the error $\tilde{\mathbf{e}}$ is obtained (Equation (4.2)).

$$p^* \triangleq \Pr[\tilde{e}_k = 1] = 2\tilde{p}(1-\tilde{p})\left(1 - \frac{w_r}{n}\right) + \left((1-\tilde{p})^2 + \tilde{p}^2\right)\frac{w_r}{n} \tag{4.2}$$

4.1.3 Decryption Failure Rate

The RMRS construction is chosen because there exists an efficiently computable upper bound for the DFR of this concatenated code. First, an upper bound for the DFR of the inner first-order RM is shown in [AMAD$^+$24, Proposition 5].

Assuming that the error to the code $\mathcal{C}$ is randomly sampled from Bernoulli$(p)^n$, the DFR of the first-order RM code $[n_{RM}, k_{RM}, d_{RM}]_2$ is denoted as p_{RM}.

$$p_{RM} \leq \sum_{l=\frac{d_{RM}}{2}}^{n_{RM}} \alpha_l p^l (1-p)^{n_{RM}-l}, \tag{4.3}$$

where

$$\alpha_l = \min\left\{\binom{n_{RM}}{l}, \frac{2^{k_{RM}}-1}{2}\left[\binom{d_{RM}}{\frac{d_{RM}}{2}}\binom{d_{RM}}{l-\frac{d_{RM}}{2}}\right.\right.$$
$$\left.\left. + 2\sum_{j=\frac{d_{RM}}{2}+1}^{d_{RM}}\binom{d_{RM}}{j}\binom{d_{RM}}{l-j} + \frac{2^{k_{RM}-1}-1}{2}\sum_{j=0}^{\frac{d_{RM}}{2}}\binom{\frac{d_{RM}}{2}}{j}\binom{\frac{d_{RM}}{2}}{l-d_{RM}+j}\right]\right\}.$$

Next, we can use p_{RM} to estimate the DFR of the overall concatenated code $\mathcal{C}$. According to [AMAD+24, Theorem 7], we can estimate the total DFR p_{RMRS} as

$$p_{RMRS} \leq \sum_{l=t_{RS}+1}^{n_{RS}}\binom{n_{RS}}{l}p_{RM}^{l}(1-p_{RM})^{n_{RS}-l}, \tag{4.4}$$

where $d_{RS} = 2t_{RS} + 1$.

Decryption failures are not desirable and can often provide an advantage to an attacker [KL07, Section 3.7]. For IND-CCA security, it is required that $p_{RMRS} \leq 2^{-\lambda}$ where λ is the desired security level. Since we approximated the error term $\tilde{\mathbf{e}}$ as a random vector sampled from Bernoulli$(p^*)^n$, we need to ensure that the RMRS code selected must be able to correct this error with high probability. In other words, we must ensure that for p^*, the DFR is low enough. A code $\mathcal{C}$ is suitable for the HQC scheme if the maximum error rate p that it can correct is at least as large as the effective noise rate p^*. That is, given a code $\mathcal{C}$, we must find

$$p = \max_{x\in[0,1]} x \quad \text{s.t.} \quad \log p_{RMRS}(x) \leq -\lambda$$

Note that this optimization problem depends only on the choice of RMRS code and length n and not on the other parameters of the scheme.

For the scheme to be able to decrypt with the desired DFR, we must ensure that –

1. $p \geq p^*$, and

2. $p_{RMRS} \leq 2^{-\lambda}$ for the security level λ.

Next, we will see how we can leverage quantization to reduce the ciphertext length while exploiting the slack between the correctable error rate p and the error rate p^* of the induced error $\tilde{\mathbf{e}}$.

4.2 Quantized HQC

Using the techniques from Chapter 3, we can apply quantization to the ciphertext $\mathbf{c}$ in the HQC cryptosystem. Our novel quantized HQC scheme is laid out in Subsection 4.2.1.

4.2.1 Our Contribution

Borrowing the strategy from Section 3.2, we will use a polar code Q to compress $\mathbf{c}$. Again, this introduces an additional quantization error, which must also be corrected using the code E. To treat the two components of the error independently, we will employ dithering. Our quantized and dithered HQC scheme is presented in Table 4.2.

Table 4.2: Quantized HQC cryptosystem with dithering.

$KeyGen(1^k)$	$Encrypt_{pk}(\mathbf{m})$	$Decrypt_{sk}(\mathbf{a}, \lfloor \mathbf{c}-\mathbf{v} \rceil_Q, \mathbf{v})$
sk: $\mathbf{s} \leftarrow \mathcal{A}_n(w_u)$	$\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3 \leftarrow \mathcal{A}_n(w_r)$	return
$\mathbf{g} \leftarrow \mathcal{R}_n$	$\mathbf{a} = \mathbf{r}_1 + \mathbf{g} \cdot \mathbf{r}_2$	$decode_{\mathcal{C}}(\lfloor \mathbf{c}-\mathbf{v} \rceil_Q + \mathbf{v} - \mathbf{a}\cdot\mathbf{s})$
$\mathbf{e} \leftarrow \mathcal{A}_n(w_u)$	$\mathbf{c} = \mathbf{b} \cdot \mathbf{r}_2 + \mathbf{r}_3 + encode_{\mathcal{C}}(\mathbf{m})$	
$\mathbf{b} = \mathbf{g} \cdot \mathbf{s} + \mathbf{e}$	$\mathbf{v} \leftarrow \mathcal{F}(Q)$	
pk: $(\mathbf{g}, \mathbf{b})$	return $(\mathbf{a}, \lfloor \mathbf{c}-\mathbf{v} \rceil_Q, \mathbf{v})$	
return pk		

As in the symmetric cryptosystem presented in Table 3.5, one need not send the entire length-n dither $\mathbf{v}$; rather, a seed of length λ would suffice. As a reminder, $\mathcal{F}(Q)$ is the fundamental region of the code Q (Definition 16), which means that the dither is picked uniformly from those vectors which decode to $\mathbf{0}$ under the polar code Q. To do this, one can randomly sample a vector from the ambient space $\tilde{\mathbf{v}} \leftarrow \mathcal{R}_n$ and then apply the decoder $\mathbf{v} = decode_Q(\tilde{\mathbf{v}})$. This ensures that $\mathbf{v} \in \mathcal{F}(Q)$ and its distribution behaves exactly like the uniform distribution on $\mathcal{F}(Q)$.

4.2.2 Correctness of Decryption

Because of the quantization, we need to include the quantization error $\{\mathbf{c}-\mathbf{v}\}_Q$ in the correctness of decryption.

$$\begin{aligned} decode_{\mathcal{C}}(\lfloor \mathbf{c}-\mathbf{v} \rceil_Q + \mathbf{v} - (\mathbf{r}_1 + \mathbf{g}\cdot\mathbf{r}_2)\cdot\mathbf{s}) &= decode_{\mathcal{C}}(\mathbf{c} + \{\mathbf{c}-\mathbf{v}\}_Q - (\mathbf{r}_1 + \mathbf{g}\cdot\mathbf{r}_2)\cdot\mathbf{s}) \\ &= \mathbf{m} + decode_{\mathcal{C}}(\tilde{\mathbf{e}} + \{\mathbf{c}-\mathbf{v}\}_Q) \end{aligned}$$

Similar to the symmetric cryptosystem, dithering removes the dependence between the quantization error $\{\mathbf{c}-\mathbf{v}\}_Q$ and the introduced error $\tilde{\mathbf{e}}$. Moreover, from [KU10, Lemma 9] we can approximate $\{\mathbf{c}-\mathbf{v}\}_Q$ with a random vector sampled from the distribution $Bernoulli(p_Q)^n$.

Now, the code $\mathcal{C}$ must be able to correct both $\tilde{\mathbf{e}}$ and $\{\mathbf{c}-\mathbf{v}\}_Q$ while maintaining DFR below $2^{-\lambda}$. The error seen by $\mathcal{C}$ is approximated as a $BSC(p)$, where

$$p = p^*(1-p_Q) + (1-p^*)p_Q$$

4.2.3 Methodology of Parameter Search

The task is to find an appropriate RMRS code $\mathcal{C}$ that can enable quantization to be applied. In other words, we need to find a code that can correct for the channel BSC(p) where $p \geq p^*$ for a fixed HQC error rate p^* (of $\tilde{\mathbf{e}}$) while maintaining a low DFR. Therefore, we first create a list of possible RMRS codes. Then, we can perform two independent steps –

1. finding the maximum p that can be corrected by a given RMRS code, and

2. selecting a prime $n \geq n_{RM}n_{RS}$ and computing the corresponding p^*.

If the $p \geq p^*$, it means that we can use the slack between them for quantization, and this code is "feasible". Then, one can compute the asymptotic compression factor $\rho = 1 - H_2(p_Q)$. Since the encryption outputs three vectors, the total ciphertext length is $n + \rho n_{RM}n_{RS} + \lambda$, where the last term is the seed length for the dither. The search strategy is summarized in Figure 4.2.3.

For the first step of creating a list of valid RMRS codes, we first begin with a list of starting RM codes of order 1. These have degrees $m \in \{6, 7, 8\}$. Then we extend this list by duplicating each code by a factor of $\{2, 3, 4, 5, 6\}$. From this starting list of RM codes, we can then find a list of RS codes that have the appropriate field size $2^{k_{RM}}$ and a minimum distance at least half that of the RS code in the current HQC specification. Now that we have successfully created a list of valid concatenated codes, we can eliminate some of them if they are too long. Concretely, if $n_{RM}n_{RS} \geq 2n_{HQC}$, where n_{HQC} is the length chosen by the HQC specification, then we have no hope of performing better than the standard. This is because the first term of the ciphertext will by itself be larger than the total ciphertext length of the current HQC standard.

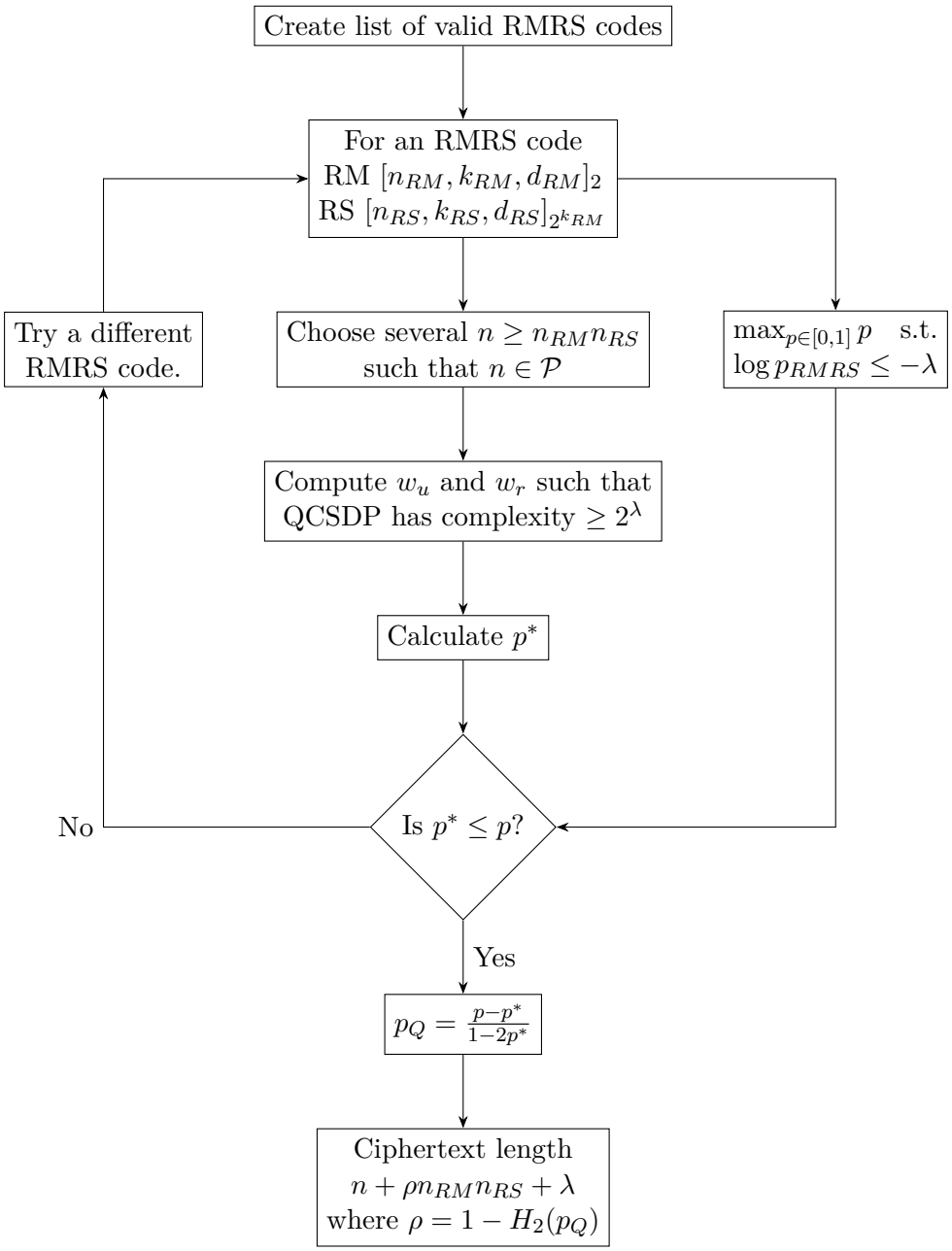


Figure 4.1: Search strategy to find optimal RMRS code for quantized HQC.

4.3 Results

We ran the search for three security levels $\lambda \in \{128, 192, 256\}$ using the above strategy in Python [VRDJ95]. Due to limitations with the maximum and minimum representable numbers using a double data type, we had to write a bisection search algorithm for multi-precision floating point numbers [mdt23].

4.3.1 Overview of the Search Results

For $\lambda = 128$, we could get a reduction of 4282 bits for the total ciphertext length, which is roughly 12 %. The details of the concatenated code chosen and the parameters are listed in Table 4.3. A comparison of the quantized and non-quantized versions in terms of the ciphertext length is presented in Table 4.4. Since some parameters of the quantized scheme differ from the non-quantized scheme, the complexity of encryption and decryption changes. This is compared in Table 4.6.

Table 4.3: Optimal parameters chosen for the quantized HQC cryptosystem along with the computed tolerable Bernoulli error rate p and its components p^* from the intentional error and p_Q from quantization.

λ	RM	RS	n	w_u	w_r	p	p^*	p_Q
128	$[512, 8, 256]_2$	$[42, 16, 27]_{2^8}$	21517	66	75	0.35523	0.30336	0.13189
192	$[768, 8, 384]_2$	$[58, 24, 35]_{2^8}$	44549	100	114	0.37445	0.32215	0.14705
256	$[768, 8, 384]_2$	$[92, 32, 61]_{2^8}$	70657	131	149	0.3843	0.33581	0.14764

Table 4.4: Comparison of ciphertext lengths between the current HQC standard and the quantized version obtained by the optimal parameters in Table 4.3 using the asymptotic rate-distortion bound Equation (2.3).

λ	HQC	Quantized HQC	Reduction (%)
128	35333	31051	12.12
192	71691	62452	12.89
256	115237	98899	14.18

For the receiver to be able to decrypt, one must send the dithering vector $\mathbf{v}$ along with the quantized ciphertext. As mentioned before, we can send a short seed of length λ instead of the full dither. The seed length must also be accounted for in the overall ciphertext length. The ciphertext length is thus estimated as $n + \rho n_{RM} n_{RS} + \lambda$, where each additive term corresponds to the components of the ciphertext $(\mathbf{a}, \lfloor \mathbf{c} - \mathbf{v} \rceil_Q, \mathbf{v})$. The estimated

Table 4.5: Comparison of ciphertext lengths between the current HQC standard and the quantized version obtained by the optimal parameters in Table 4.3 for the finite-length rate-distortion bound Equation (2.4).

λ	HQC	Quantized HQC	Reduction (%)
128	35333	31066	12.08
192	71691	62467	12.87
256	115237	98916	14.16

Table 4.6: Comparison of public-key lengths between the current HQC standard and the quantized version obtained by the optimal parameters in Table 4.3.

λ	HQC	Quantized HQC	Increase (%)
128	17797	21645	21.62
192	36043	44741	24.13
256	57893	70913	22.49

ciphertext lengths provided in Table 4.4 use the asymptotic rate-distortion bound Equation (2.3). In Table 4.5, we use the finite-length rate-distortion bound (Equation (2.4)) from [KV12].

4.3.2 Interpretation and Impact Analysis

From Table 4.4, it is clear that quantization can indeed reduce the length of the overall ciphertext. However, it comes at a cost – the complexity of encryption and decryption is increased due to the larger n value. Notice that the length n is significantly higher in the quantized version as compared to the standard HQC parameter set. This increases the complexity of the key generation, $encode_{\mathcal{C}}(\cdot)$ and $decode_{\mathcal{C}}(\cdot)$ steps. The polynomial multiplications used in these steps have a time complexity in the order of $O(n^{1.59})$ using Karatsuba's algorithm [KO63]. Hence, an increase in n negatively affects the encryption and decryption time.

Further, the length of the public key is also increased noticeably, as seen in Table 4.6. Since the public key consists of two vectors from $\mathcal{R}_n$, its length can be assumed to be $2n$. However, this can be reduced further by using a seed for the first component $\mathbf{g}$. This is a random public generator polynomial. Hence, we can perform this optimization. In our comparison in Table 4.6, we have assumed a public-key length of $n + \lambda$.

Further, for encryption, one needs to use a decoder for the polar code Q in order to quantize the ciphertext. This adds an additional cost of $O(n \log n)$ if one chooses to

use the (sub-optimal) successive cancellation decoder. For a compression factor closer to the rate-distortion bound Equation (2.3), a successive cancellation list decoder can be used, but this adds even more complexity [TV15]. In the decryption step, an overhead of $O(n \log n)$ is added due to the polar encoder used in the de-quantization or reconstruction step. Thus, we are trading off the time complexity of encryption and decryption for more efficient communication.

In summary, we have showcased the advantages of using quantization on the HQC cryptosystem to reduce the ciphertext length. A maximum reduction of 14.18 % was seen for $\lambda = 256$ and the smallest reduction of 12.12 % was observed in the case of $\lambda = 128$. These are promising results that indicate that, indeed, quantization can be a valuable tool in improving the ciphertext lengths of post-quantum cryptography.

5 Conclusion

In this work, we analyzed a code-based symmetric cryptosystem with vector quantization and also applied this technique to reduce the size of the HQC ciphertext by 12% without affecting the security. Although the reduction is moderate, it provides a tangible step toward addressing one of the key limitations of code-based cryptosystems. The Hamming Quasi-Cyclic scheme has suffered from long ciphertext lengths. With quantization, one can mitigate this disadvantage without adding a lot to the complexity of the encryption and decryption. We also observed that quantization is more effective as the ciphertext length increases because practical lossy source coders approach the rate distortion bound asymptotically. Polar codes can play a key role, as they provide asymptotic guarantees while maintaining low complexity. While HQC is an ideal candidate for quantization due to its large ciphertext length and adaptability to correct the quantization errors, other schemes like the Niederreiter and McEliece cryptosystems are more difficult to improve. A potential direction for future work could be to explore other code constructions in HQC, which may perform better error correction, decreasing the theoretical bound of compression even further. Due to the concatenated RMRS code construction in HQC, one could choose to quantize block-wise on each RM codeword instead of the entire codeword. Although the compression factor would not be close to the rate-distortion limit, it creates a more structured quantization error, which can be taken advantage of during decryption. It would also be interesting to study the application of quantization to the McEliece and Niederreiter cryptosystems while keeping the security level undisturbed. So far, we have only considered binary codes; it is possible that when working over larger fields, quantization is more effective; however, it comes with increased complexity. A detailed analysis of the correctness of decryption in the symmetric cryptosystem could potentially help understand the asymmetric systems better.

Bibliography

[ABD+20] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "CRYSTALS–kyber: Algorithm specification and supporting documentation (version 3.02)," Round-3 submission to the NIST Post-Quantum Cryptography Standardization Project, 2020, https://pq-crystals.org/kyber/data/kyber-specification-round3-20210804.pdf.

[AMAB+25] C. Aguilar-Melchor, N. Aragon, S. Bettaieb, L. Bidoux, O. Blazy, J. Bos, J.-C. Deneuville, A. Dion, P. Gaborit, J. Lacan, E. Persichetti, J.-M. Robert, P. Véron, and G. Zémor, "Hamming quasi-cyclic (HQC) specification – fourth round version," HQC Team, Tech. Rep., February 2025. [Online]. Available: https://pqc-hqc.org/doc/hqc-specification_2025-02-19.pdf

[AMAD+24] C. Aguilar-Melchor, N. Aragon, J.-C. Deneuville, P. Gaborit, J. Lacan, and G. Zémor, "Efficient error-correcting codes for the hqc post-quantum cryptosystem," *Designs, Codes and Cryptography*, vol. 92, no. 12, pp. 4511–4530, Dec 2024. [Online]. Available: https://doi.org/10.1007/s10623-024-01507-6

[AMBD+18] C. Aguilar-Melchor, O. Blazy, J.-C. Deneuville, P. Gaborit, and G. Zémor, "Efficient encryption from random quasi-cyclic codes," *IEEE Transactions on Information Theory*, vol. 64, no. 5, pp. 3927–3943, 2018.

[Ari09] E. Arikan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051–3073, 2009.

[bacr25] s. s. bhoi, a. arakala, a. b. corman, and a. rao, "post-quantum homomorphic encryption: a case for code-based alternatives," *cryptography*, vol. 9, no. 2, 2025. [Online]. Available: https://www.mdpi.com/2410-387x/9/2/31

[BMvT78] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *IEEE Transactions on Information Theory*, vol. 24, no. 3, pp. 384–386, 1978.

[Cov99] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[CTS16] R. Canto Torres and N. Sendrier, "Analysis of information set decoding for a sub-linear error weight," in *Post-Quantum Cryptography*, T. Takagi, Ed. Cham: Springer International Publishing, 2016, pp. 144–161.

[Gro96] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, ser. STOC '96. New York, NY, USA: Association for Computing Machinery, 1996, p. 212–219. [Online]. Available: https://doi.org/10.1145/237814.237866

[KL07] J. Katz and Y. Lindell, *Introduction to modern cryptography: principles and protocols*. Chapman and hall/CRC, 2007.

[KO63] A. Karatsuba and Y. Ofman, "Multiplication of Multidigit Numbers on Automata," *Soviet Physics Doklady*, vol. 7, p. 595, Jan. 1963.

[KU10] S. B. Korada and R. L. Urbanke, "Polar codes are optimal for lossy source coding," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1751–1768, 2010.

[KV12] V. Kostina and S. Verdu, "Fixed-length lossy compression in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 58, no. 6, p. 3309–3338, Jun. 2012. [Online]. Available: http://dx.doi.org/10.1109/TIT.2012.2186786

[LLL24] S. Lyu, L. Liu, and C. Ling, "Learning with quantization: Construction, hardness, and applications," Cryptology ePrint Archive, Paper 2024/714, 2024. [Online]. Available: https://eprint.iacr.org/2024/714

[McE78] R. J. McEliece, "A public-key cryptosystem based on algebraic," *Coding Thv*, vol. 4244, no. 1978, pp. 114–116, 1978.

[mdt23] T. mpmath development team, *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 1.3.0)*, 2023, http://mpmath.org/.

[MS23] D. Micciancio and M. Schultz, "Error correction and ciphertext quantization in lattice cryptography," Cryptology ePrint Archive, Paper 2023/525, 2023. [Online]. Available: https://eprint.iacr.org/2023/525

[Nie86] H. Niederreiter, "Knapsack-type cryptosystems and algebraic coding theory," *Prob. Contr. Inform. Theory*, vol. 15, no. 2, pp. 157–166, 1986.

[Pie12] K. Pietrzak, "Cryptography from learning parity with noise," in *SOFSEM 2012: Theory and Practice of Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 99–114.

[Ran25] H. Randriambololona, "The syzygy distinguisher," in *Advances in Cryptology – EUROCRYPT 2025*, S. Fehr and P.-A. Fouque, Eds. Cham: Springer Nature Switzerland, 2025, pp. 324–354.

[Reg09] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, Sep. 2009. [Online]. Available: https://doi.org/10.1145/1568318.1568324

[Rot06] R. Roth, *Introduction to Coding Theory*. Cambridge University Press, 2006.

[Sha48] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[Sho99] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999. [Online]. Available: https://doi.org/10.1137/S0036144598347011

[Sip13] M. Sipser, *Introduction to the Theory of Computation*, 3rd ed. Boston, MA: Course Technology, 2013.

[Sta23] W. Stallings, "Cryptography and network security," 2023.

[TV15] I. Tal and A. Vardy, "List decoding of polar codes," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2213–2226, 2015.

[VRDJ95] G. Van Rossum and F. L. Drake Jr, *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[YZW+17] Y. Yu, J. Zhang, J. Weng, C. Guo, and X. Li, "Collision resistant hashing from sub-exponential learning parity with noise," Cryptology ePrint Archive, Paper 2017/1260, 2017. [Online]. Available: https://eprint.iacr.org/2017/1260