Code-based cryptography over the Lee Metric

Mathias Marty

School of Computer and Communication Sciences

Semester Project

March 2023

Responsible
Prof. Serge Vaudenay
EPFL / LASEC

Supervisor
Mr. Bénédikt Tran
EPFL / LASEC

Abstract. In 2020, Horlemann-Trautmann and Weger argued that by considering the McEliece cryptosystem over $\mathbb{Z}/p^m\mathbb{Z}$ embedded with the Lee metric, the size of the public key can be drastically reduced [47]. This is significant because the McEliece cryptosystem is one of the four finalists in the NIST call for proposals for post-quantum PKE/KEM [1] and its public key size is exactly its main drawback. Their conclusion about potential key size reduction is motivated by the fact that the best known generic decoding algorithms, the ISD family, are severely affected when adapted over $\mathbb{Z}/p^m\mathbb{Z}$ equipped with the Lee metric [19]. On the other hand, in 2001, Al Jabri proposed another decoding algorithm called statistical decoding [20]. This algorithm and its best variants have long been forgotten because they were largely outperformed by ISD algorithms. However, it would be interesting to analyze how statistical decoding algorithms perform over $\mathbb{Z}/p^s\mathbb{Z}$ in the Lee metric, given the growing interest in code-based cryptography over rings in the Lee metric. Furthermore, Niebuhr proved in 2011 that for q large enough, statistical decoding outperforms the best ISD algorithms over $\mathbb{F}_q$ in the Hamming metric [31]. This motivates our interest in translating statistical decoding over $\mathbb{Z}/p^m\mathbb{Z}$ into the Lee metric. To the best of our knowledge, there is no generalization of statistical decoding over such mathematical structures. This thesis aims to fill this gap and provide the necessary tools to compare both ISD and statistical decoding algorithms. We prove that the Niebuhr observation holds over $\mathbb{Z}/p^s\mathbb{Z}$ in the Lee metric. Finally, we discuss how we may improve our generalized statistical decoding algorithm to fix the public key size in a concrete cryptographic setting.

Table of Contents

1 Introduction

In an increasingly interconnected world, where information flows seamlessly across networks and digital systems, the security of our data is of great importance. The field of cryptography has long been at the forefront of protecting sensitive information. However, as technology evolves, traditional cryptographic algorithms face an imminent threat from the rise of quantum computing.

To safeguard against the impending threat, researchers and cryptographic experts have been exploring a new kind of cryptographic scheme known as *post-quantum cryptography*. These schemes aim to develop public key cryptosystems that are resistant to attacks by both classical and quantum computers. Their security relies on mathematical problems that are not believed to be breakable by quantum computers

The National Institute of Standards and Technology (NIST) launched in 2016 a call for proposals for Post-Quantum (PQ) cryptography scheme [1]. Among the seven still in the running cryptosystems is the *McEliece Public-Key Encryption (PKE)* which is based on code-based cryptography [28]. Code-based cryptography leverages error-correcting codes, which have been extensively studied and widely used in various communication systems, to construct cryptographic primitives that are resistant to quantum attacks. By utilizing the properties of error-correcting codes, such as their ability to correct errors and provide information-theoretic security, code-based cryptography presents a promising avenue for securing our digital infrastructure against the threat of quantum computing.

Despite its strengths, the McEliece cryptosystem is not without its drawbacks. One significant disadvantage is its relatively large key size compared to other post-quantum and classical encryption schemes like RSA or Elliptic Curve Cryptography (ECC). This large key size presents challenges in terms of storage, transmission, and computational overhead.

The size of the public keys is set so that the best-known algorithms need more computation time than the workfactor threshold defined by the security parameter. Obviously, there is no way one could certify whether our current algorithms are the best. Regarding code-based cryptography, the two most important families of algorithms are *Information Set Decoding (ISD)* and *statistical decoding*. Over the classical binary field, it seems that the former performs better, and thus statistical decoding has been unpopular during the last period [9].

The McEliece cryptosystem was first proposed over the binary fields equipped with the Hamming metric. Recent works by Violetta Weger *et al.* showed that changing the ambient space to a prime power residue ring and swapping the metric for the *Lee metric* may drastically reduce the public key size [47]. Their argument is based on the fact that the well knows ISD algorithms perform worth when adapted to the Lee metric. However, we could also consider translating the statistical decoding algorithm into the Lee metric. As we will see, it does not suffer as much as ISD algorithms under this transformation.

This thesis studies how the generalized statistical decoding algorithm performs compared to ISD algorithms over integer residue ring embedded with the Lee metric. We also briefly restate some important ISD algorithms together with

their complexity. A more in-depth explanation of this family of algorithms and how to extend them into the Lee metric can be found in Violleta Weger's thesis [45].

1.1 Contribution

We show in Section 3 that we can naturally extend the statistical decoding algorithm over $\mathbb{Z}/m\mathbb{Z}$ equipped with the Lee metric. We give expressions to compute its running time and show that it is not negatively affected by the metric change and by the weaker structure of a finite commutative ring with identity compared to that of a finite field.

We prove in Section 5 that, for sufficiently large rings, statistical decoding outperforms BJMM even when the precomputation step is included in the running time. Finally, we discuss in Section 6 possible improvements to statistical decoding over the Lee metric. We hope that this last section will motivate further research in this area.

1.2 Organization

The first part of this thesis introduces the mathematical background necessary to understand and generalize both ISD and statistical decoding. First, we introduce notions of linear algebra and then continue with linear codes. Finally, we bring in the formal definition of a public key cryptosystem and the definitions of the McEliece and Lee-McEliece cryptosystems together with some formal statements about their security.

Section 3 focuses on statistical decoding. We first introduce statistical decoding over finite fields embedded with the Hamming metric and then, generalize the algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric. An expression for its complexity is given so that we can compare it with other algorithms.

Another family of decoding algorithms, called ISD is presented in Section 4. We present some important algorithms of this family and how they work over the Lee metric. Expressions for their complexity are also given.

Section 5 compares statistical decoding with ISD algorithms and discusses our results. Finally, we will review our work and discuss how we could improve it in Section 6.

2 Background

2.1 Notation

Sets, Rings, and Fields Throughout this thesis, we will denote by $\mathbb{F}_q$ the finite field of cardinality q, $\mathbb{Z}/n\mathbb{Z}$ the ring of integers modulo n and $\mathbb{N}$ the set of positive integers including zero. Note that, as a consequence of field theory, q must be a prime power. For any integers a and b such that $a < b$, the set $\{a, \dots, b\}$ is symbolized with $[\![a, b]\!]$ and $\{s+c \mid s \in S\}$ with $S+c$. By $\mathcal{R}^\times$ we will represent the

unit group of the ring $\mathcal{R}$. We will simplify the notation $[\![1, a]\!]$ with $[\![a]\!]$. Finally, we will denote the n-wise external direct sum of S with itself by S^n and by internal direct sum by $\oplus$.

Linear Algebra Vectors will be represented with bold symbols (*i.e.*, $\mathbf{x}$) and their components, as indexed symbols (*i.e.*, $\mathbf{x}_i$). They will be considered as row vectors (*i.e.* $[1, 2, 3]$). We will denote by $S \leq V$ and $S < V$ the fact that S is respectively a linear subspace and a proper linear subspace of V. The same notation is used for modules and their submodules. In the event that the meaning of this symbol could be ambiguous, we will explicitly state to which structure it applies. Let S be any set, $\mathbf{G} \in S^{m \times n}$, $X \subset [\![m]\!]$ and $Y \subset [\![n]\!]$. We will denote by $\mathbf{G}^{\mathfrak{R}(X)}$ and $\mathbf{G}^{\mathfrak{C}(Y)}$ the respective matrices composed with the rows and colomns of G indexed by X and Y. When $\mathbf{G}$ is a one-dimensional row matrix, also known as a vector, we will simply write $\mathbf{G}_Y$ instead of the cumbersome $\mathbf{G}^{\mathfrak{C}(Y)}$ notation. We will write $\mathbf{0}$ and $\mathbf{1}$ the respective zero and one vector/matrix. Their sizes will be either clear from the context or specified on the index (*i.e.* $\mathbf{0}_{n \times m}$). The symbol $\mathbf{I_k}$ will denote the identity matrix of size k. The next notation will be useful when describing algorithms that work on a particular set of indices of a vector. For any $X \subset [\![n]\!]$ of size m and set S, we will denote by $\sigma_X \colon S^m \to S^n$ the respective canonical embedding.

2.2 Hamming and Lee metric

In this section, we present the *metrics* used in this thesis. For the following definitions, we let n be a strictly positive integer and $\mathcal{R}$ be a finite commutative ring with identity of cardinality q. For the sake of clarity, we will refer to finite commutative rings with identity simply as rings.

Definition 1 (Hamming and Lee Distances). *For any $\mathbf{x}, \mathbf{y} \in \mathcal{R}^n$, their Hamming and Lee distances, denoted by $d_H(\mathbf{x})$ and $d_L(\mathbf{x})$, respectively, are defined to be:*

$$d_H(\mathbf{x}, \mathbf{y}) = |\{i \in [\![n]\!] \mid \mathbf{x}_i \neq \mathbf{y}_i\}|,$$
$$d_L(\mathbf{x}, \mathbf{y}) = \sum_{i \in [\![n]\!]} \min(|\mathbf{x}_i - \mathbf{y}_i|, q - |\mathbf{x}_i - \mathbf{y}_i|).$$

Note that any metric induces a *weight function* by taking the distance to the vector $\mathbf{0}$. This motivates the following two weight functions:

Definition 2 (Hamming and Lee Weights). *For any $\mathbf{x} \in \mathcal{R}^n$, its Hamming and Lee weights, denoted by $wt_H(\mathbf{x})$ and $wt_L(\mathbf{x})$, respectively, are defined to be:*

$$wt_H(\mathbf{x}) = |\{i \in [\![n]\!] \mid \mathbf{x}_i \neq 0\}|,$$
$$wt_L(\mathbf{x}) = \sum_{i \in [\![n]\!]} \min(\mathbf{x}_i, q - \mathbf{x}_i).$$

Remark 1. In the binary case, this corresponds to summing up all the bits, *i.e.*, counting the number of non-zero entries.

Remark 2. Consider the Lee weight of any $\mathbf{x} \in (\mathbb{Z}/2\mathbb{Z})^n$,

$$\begin{aligned} \mathrm{wt}_L(\mathbf{x}) &= \sum_{i \in [\![n]\!]} \min(\mathbf{x}_i, 2 - \mathbf{x}_i) \\ &= \sum_{i \in [\![n]\!]} \mathbf{x}_i && \text{By Remark 1.} \\ &= \mathrm{wt}_H(\mathbf{x}). \end{aligned}$$

In other words, over $(\mathbb{Z}/2\mathbb{Z})^n$, the Lee and the Hamming metrics are identical. A similar proof exists for $(\mathbb{Z}/3\mathbb{Z})^n$. The Lee weight will therefore only be of interest in non-binary and non-ternary cases.

We introduce the notion of *spheres* and *balls*. We will use them regularly in this thesis.

Definition 3 (Balls and Spheres). *Let $q = |\mathcal{R}|$. The ring $\mathcal{R}$ will either be clear from the context or explicitly specified. We define a Hamming ball and a Hamming sphere centered at $\mathbf{x}$ of radius r respectively as follows:*

$$B^n_{q,H}(\mathbf{x}, r) = \{\mathbf{y} \in \mathcal{R}^n \mid d_H(\mathbf{y}, \mathbf{x}) \leq r\}.$$

Similarly, we have for the Lee metric:

$$B^n_{q,L}(\mathbf{x}, r) = \{\mathbf{y} \in \mathcal{R}^n \mid d_L(\mathbf{y}, \mathbf{x}) \leq r\}.$$

When also defines the notion of a sphere:

$$S^n_{q,H}(\mathbf{x}, r) = \{\mathbf{y} \in \mathcal{R}^n \mid d_H(\mathbf{y}, \mathbf{x}) = r\}.$$

This definition applies similarly to the Lee metric.

The Hamming ball has the following cardinality:

Proposition 1 (Cardinality of the Hamming Ball). *The cardinality of the Hamming ball is easily seen to be:*

$$|B^n_{q,H}(\mathbf{x}, r)| = \sum_{i=0}^{r} \binom{n}{i} (|\mathcal{R}| - 1)^i. \tag{1}$$

Note that this value does not depend on the center point $\mathbf{x}$.

The complexity analysis of the ISD algorithms over the Lee metric requires an expression for the cardinality of a Lee sphere. The problem is more technical than in the Hamming case, an approximation is presented in [47] using generating functions. The exact formula is given in [45]. Before presenting this result, we have to introduce the following function:

Definition 4. *Let* $f(n, \ell, w, p^s) = |\{\mathbf{x} \in (\mathbb{Z}/p^s\mathbb{Z})^n \mid wt_H(\mathbf{x}) = \ell \wedge wt_L(\mathbf{x}) = w\}|$.

A closed expression for this function is given by the following proposition:

Proposition 2 ([45, Proposition 5.1.9]). *Let* n, ℓ *and* w *be positive integers that respect* $1 \leq w \leq \lfloor \frac{p^s}{2} \rfloor$ *and* $\ell \leq \min\{n, w\}$. *We have that,*

$$f(n, \ell, w, p^s) = \begin{cases} \sum_{k=0}^{\min\{\ell, \lfloor \frac{2w}{p^s} \rfloor\}} \binom{n-k}{\ell-k} 2^{\ell-k} \binom{n}{k} C(w - k\frac{p^s}{2}, \ell - k, \frac{p^s}{2} - 1) & \text{if } p^s \text{ is even,} \\ \binom{n}{\ell} 2^\ell C(w, \ell, \lfloor \frac{2^s}{2} \rfloor) & \text{if } p^s \text{ is odd.} \end{cases}.$$

The function $C(w, \ell, \mu)$ counts the number of ways we can partition w with ℓ parts such that each part is at most μ. This function will also be of interest when extending statistical decoding to the Lee metric in Section 3. We now introduce a closed formula for $C(w, \ell, \mu)$.

Proposition 3 (Compositions of an Integer with Restricted Parts [2]). *Let* n, ℓ *and* w *be positive integers such that* $1 \leq \ell$ *and* $\mu \leq w$, *then*

$$C(w, \ell, \mu) = \sum_{j=0}^{\min\{\ell, \lfloor \frac{w-\ell}{\mu} \rfloor\}} (-1)^j \binom{\ell}{j} \binom{w - j\mu - 1}{\ell - 1}.$$

We use the convention $C(0, 0, \mu) = 1$

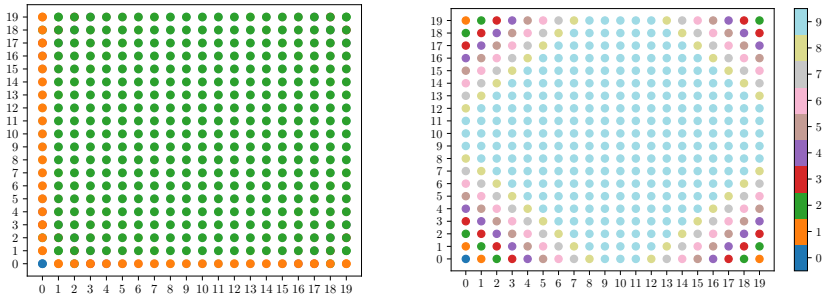
The formula for the cardinality of a Lee sphere can finally be introduced.

Theorem 1 (Cardinality of a Lee Sphere [45, . Corollary 5.1]). *For any positive integers* n, w *and* s *such that* $0 \leq w \leq \lfloor \frac{p^s}{s} \rfloor$, *the cardinality of the Lee sphere of radius* w *over* $(\mathbb{Z}/p^s\mathbb{Z})^n$ *is*

$$F(n, w, p^s) = \sum_{\ell=1}^{\min\{n,w\}} f(n, \ell, w, p^s).$$

Observe that the cardinality of a ball or a sphere does not depend on its center point $\mathbf{x}$.

To get an intuitive sense of how the metrics act on a two-dimensional space, we show in Figure 1 the respective Lee weights and Hamming weights of all elements in the ring $(\mathbb{Z}/20\mathbb{Z})^2$. Note that in the Hamming case (1a), the two orange lines correspond to all vectors that are one coordinate different from the vector $\mathbf{0}$. In the Lee case (1b), the weight is not as intuitive. Some low-weight elements appear in the corner of the space, which is a result of the min term in the metric's expression. It collapses the corner of the space, and the highest-weight elements appear in the center.



(a) Hamming weight. (b) Lee weight.

Fig. 1: Comparison of the Hamming and Lee weights of the elements of the ring $(\mathbb{Z}/20\mathbb{Z})^2$.

2.3 Linear and Ring Linear Codes

In this section, we introduce the concept of a *linear code*. This notion is the cornerstone of code-based cryptography, and we must therefore define it properly. Throughout this section, we let m and n be strictly positive integers and q be a prime power. Linear codes can be defined over fields or, more generally, over finite commutative rings. In the case of fields, we will consider the $\mathbb{F}_q$*-space* $\mathbb{F}_q^n$, while in the ring case, we will restrict ourselves to the $\mathbb{Z}/m\mathbb{Z}$*-module* $(\mathbb{Z}/m\mathbb{Z})^n$. We first introduce the notions of linear and ring-linear code and then endow these structures with a metric. We will use the Lee metric over the rings, whereas the Hamming metric will be considered over fields. For a thorough explanation of ring-linear codes, the reader might consult at [14].

Definition 5 (Linear Codes). *A linear code $\mathcal{C}$ over $\mathbb{F}_q$ is a linear subspace of $\mathbb{F}_q^n$, using our notation, $\mathcal{C} \leq \mathbb{F}_q^n$. We refer to $k \triangleq \mathcal{C}$ as the dimension of the code, and we say that $\mathcal{C}$ is a $[n,k]$-code or a $[n,k]_q$-code if we want to specify the size of the field.*

We can generalize this definition to $\mathbb{Z}/m\mathbb{Z}$-modules.

Definition 6 (Ring-Linear codes). *A ring-linear code $\mathcal{C}$ over $\mathbb{Z}/m\mathbb{Z}$ is a submodule of $(\mathbb{Z}/m\mathbb{Z})^n$, using our notation, $\mathcal{C} \leq (\mathbb{Z}/m\mathbb{Z})^n$. We say that $\mathcal{C}$ has length n, type $h \triangleq |\mathcal{C}|$ and is a $\mathbb{Z}/m\mathbb{Z}$-code.*

Remark 3. We say that a module is k*-generated* if there exists an independent set of size k that spans the module. The size of the minimum independent set can be thought of as the "dimension" of the module, but there may be no such independent sets, as shown in the following example. Since the idea of dimension does not make sense in modules, we have introduced the notion of a *type*. For a comprehensive introduction to modules, the reader may consult [38, Chapter 4].

Example 1 (A Non Free Submodule). For example, consider the $\mathbb{Z}$-submodule $\mathcal{M} = (\mathbb{Z}/m\mathbb{Z}) \oplus \{0\}^{n-1} < (\mathbb{Z}/m\mathbb{Z})^n$. Choose any singleton set $\{(x, \mathbf{0})\}$ with $(x, \mathbf{0}) \in \mathcal{M}$. Then, $n(x, \mathbf{0}) = (nx, \mathbf{0}) = (0, \mathbf{0}) = \mathbf{0}$. Therefore, $\mathcal{M}$ has no linearly independent set and thus, no "basis". In module theory, we say that the submodule is not *free*.

On the contrary, we show that the module of interest, namely the $\mathbb{Z}/m\mathbb{Z}$-module $(\mathbb{Z}/m\mathbb{Z})^n$, is free.

Theorem 2 (The $\mathbb{Z}/m\mathbb{Z}$-Module $(\mathbb{Z}/m\mathbb{Z})^n$ is Free). *Let m and n be strictly positive integers, it follows that the $\mathbb{Z}/m\mathbb{Z}$-module $(\mathbb{Z}/m\mathbb{Z})^n$ is free.*

Proof. The module is generated by the set

$$\{\underbrace{(1, 0, \ldots, 0)}_{n \text{ times}}, \underbrace{(0, 1, 0, \ldots, 0)}_{n \text{ times}}, \ldots, \underbrace{(0, 0, \ldots, 0, 1)}_{n \text{ times}}\}$$

of size n. Note that this spanning set is independent, which concludes the proof. □

Let $\mathcal{C} \leq \mathcal{X}$ be any code over $\mathcal{X}$, where $\mathcal{X}$ is either a commutative finite ring or a finite field. The elements of the substructure $\mathcal{C}$ are called *codewords* while those of $\mathcal{X} \setminus \mathcal{C}$ are called *erroneous codewords*. The former represents elements that are not affected by any errors. An important question in coding theory is: What is the minimum distance between two codewords? This is called the minimum distance of the code. The distance functions we will use are the metrics defined in Section 2.3, *i.e.*, the Hamming, and the Lee metrics. Therefore, we will endow the ambient spaces with these two metrics. Note that due to the linearity of the code, finding the minimum distance is equivalent to finding the minimum weight of any codewords. This motivates the two following definitions. Let $\mathcal{C}$ be a linear or a ring linear code.

Definition 7 (Minimum Distance). *The minimum Hamming distance, denoted by $d_H(\mathcal{C})$, and minimum Lee distance, denoted by $d_L(\mathcal{C})$, of the code $\mathcal{C}$ are defined to be:*

$$d_H(\mathcal{C}) = \min\{wt_H(\mathbf{x}) \mid \mathbf{x} \in \mathcal{C} \setminus \{\mathbf{0}\}\},$$
$$d_L(\mathcal{C}) = \min\{wt_L(\mathbf{x}) \mid \mathbf{x} \in \mathcal{C} \setminus \{\mathbf{0}\}\}.$$

These values are directly related to the number of errors we can correct and detect, as shown in the following results:

Proposition 4 (Error Capability). *Let $\mathcal{C}$ be a code with distance d. It can detect $d-1$ errors and correct $\lfloor \frac{d-1}{2} \rfloor$ errors.*

In the case of linear code over a field, we use the following common notation that includes the minimum distance of a code:

Notation 1 *Let $\mathcal{C}$ be a $[n, k]$-code with distance d. We say that $\mathcal{C}$ is a $[n, k, d]$-code or a $[n, k, d]_q$-code if we want to specify the field size.*

To get an intuitive idea of how the minimum distance relates to the error capability, each codeword $\mathbf{x}$ can be seen as the center of the ball of radius $\frac{d-1}{2}$, where d is the minimal distance of the code. The set of vectors in this ball is the set of elements that can be correctly decoded since their closest codeword is always $\mathbf{x}$. It is obvious that the balls depend on the metric. Figure 1 gives an idea about the aspect of the balls $B^n_{q,H}(\mathbf{x}, \frac{d-1}{2})$ and $B^n_{q,L}(\mathbf{x}, \frac{d-1}{2})$.

Given n and k, one could ask what is the best code we could get. The following well-known result is an upper bound on the minimal distance, called the *singleton bound*. We state it in the case of fields and $\mathbb{Z}/m\mathbb{Z}$-codes.

Theorem 3 (Singleton Bound [40, Theorem 2]). *For any $[n, k, d]$-code, we have $k \leq n - d + 1$. For any ring linear code $\mathcal{C}$ with distance d and length n, we have $d \leq \lfloor \frac{m}{2} \rfloor (n + \log_m(|\mathcal{C}|) + 1)$.*

Another important result is the Gilbert-Varshamov bound. It shows that for any integer d, there exists some parameter such that there is a code of minimum Hamming distance of at least d. We present this theorem over finite fields and integer residue rings embedded with the Hamming and Lee metrics. The result over $\mathbb{F}_q$ equipped with the Hamming metric is taken from [39, Theorem 4.4], the one over $\mathbb{F}_p$ equipped with the Lee metric is given in [39, Theorem 10.12] and finally, the result for $\mathbb{Z}/p^m\mathbb{Z}$ over the Lee metric is from [6, Theorem 13.73].

Theorem 4 (Gilbert-Varshamov Bound [39,6]). *Let q be a prime power, $k \leq n$ and d' be positive integers. If*

$$|B^{n-1}_{q,H}(\mathbf{0}, d' - 2)| < q^{n-k},$$

then there exists a $[n, k, d]_q$-code over the Hamming metric with $d \geq d'$.

Let p be an odd prime, $k \leq n$ and d' be positive integers, then if

$$\frac{|B^{n-1}_{q,L}(\mathbf{0}, d' - 2)| - 1}{2} < \frac{p^{n-k+1} - 1}{p - 1},$$

there exists $[n, k, d]_q$-code over the Lee distance with $d \geq d'$.

Finally, let n and d' be any positive integer. There exists a code $\mathcal{C}$ over $\mathbb{Z}/p^m\mathbb{Z}$ embedded with the Lee metric such that,

$$\begin{cases} |\mathcal{C}| < \frac{p^{mn}}{(|B^{n-1}_{q,L}(\mathbf{0}, d'-2)|-1)(p^m-1)} & \text{if } p = 2 \\ |\mathcal{C}| < \frac{p^{mn}}{((|B^{n-1}_{q,L}(\mathbf{0}, d'-2)|-1)/2+1)(p^m-1)} & \text{if } p \neq 2. \end{cases}$$

Note that in the last part, the implication is in the other direction.

We bring in the important notion of *dual code*. It is related to the *parity check matrix* that we introduce later and that we will use to generate parity check equations in order to decode an erroneous codeword. For the sake of simplicity, we will only consider rings of the form $\mathbb{Z}/p^m\mathbb{Z}$. This is not restrictive since, as we will see in Section 2.4, we will be interested in cryptosystems over this type of ring. Below, the results and definitions are expressed over $\mathbb{F}_q$ but they apply in the same way over $\mathbb{Z}/p^m\mathbb{Z}$.

Definition 8 (Dual Code). *The dual code of $\mathcal{C}$, denoted by $\mathcal{C}^\perp$, is defined as:*

$$\mathcal{C}^\perp = \{\mathbf{x} \in \mathbb{F}_q^n \mid \forall \mathbf{y} \in \mathcal{C} \quad \langle \mathbf{x}, \mathbf{y} \rangle = 0\}.$$

Now we explain how one can encode and check for membership in the code using matrices.

Definition 9 (Generator Matrix). *Given $\mathbf{G} \in \mathbb{F}_q^{k \times n}$, if $\mathcal{C} = \{\mathbf{xG} \mid \mathbf{x} \in \mathbb{F}_q^k\}$, we say that $\mathbf{G}$ is a generator matrix of the code $\mathcal{C}$.*

Therefore, multiplying vectors by a matrix is a natural way to encode a message. We now present how to verify that a given element of the ambient space $\mathbb{F}_q^n$ is a codeword.

Definition 10 (Parity Check Matrix). *A parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ is a generator of the dual code $\mathcal{C}^\perp$. This means that $\mathcal{C} = \{\mathbf{y} \in \mathbb{F}_q^n \mid \mathbf{yH}^\top = \mathbf{0}\}$*

Let $\mathbf{y}$ be a codeword, $\mathbf{e}$ be a error vector and $\mathbf{y}' = \mathbf{y} + \mathbf{e}$. By linearity, we have that $\mathbf{yH}^\top = (\mathbf{y} + \mathbf{e})\mathbf{H}^\top = \mathbf{xH}^\top + \mathbf{eH}^\top = \mathbf{eH}^\top = \mathbf{s}$, where $\mathbf{s}$ is called the *syndrome* of $\mathbf{y}$. Note that the syndrome of $\mathbf{y}$ depends only on $\mathbf{e}$. The indices of the non-zeroes elements of $\mathbf{e}$ have a special name.

Definition 11 (Support of a Vector). *Let $\mathcal{R}$ be any ring. The support of a vector $\mathbf{x} \in \mathcal{R}^n$, denoted by* $\mathrm{Supp}(x)$*, is defined to be*

$$\{i \in [\![n]\!] \mid \mathbf{x}_i \neq 0\}.$$

There exists a specific type of code that provide a mechanical way of constructing the parity check matrix from the generator one, they are called *systematic code*. To introduce them, we must first explain the concept of *codes equivalence*.

Definition 12 (Code Equivalence). *Two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ are equivalent if and only if there is an isomorphism (of modules or vector spaces) between them.*

Under this relation, codes, and thus their generators, form a set of equivalence classes. An interesting representative of these classes is the systematic code. We can think of the latter as the code that rearranges any codeword $\mathbf{xG}$ so it is of the form,

$$(\mathbf{x}, r_1, \ldots, r_{n-k}), \quad \mathbf{x} \in \mathbb{F}_q^n, r_i \in \mathbb{F}_q, \text{ for } i \in [\![n-k]\!]. \tag{2}$$

In other words, it associates any messages with a codeword whose k first elements contain the message in clear. Systematic forms are used in practice as they provide direct access to the plain message. In general, the set of indices of the element that contains the plaintext message is called the *information set*, and we will denote it by $\mathcal{I}(\mathcal{C})$.

Notation 2 (Information Set) *We symbolize by $\mathcal{I}(\mathcal{C})$, the information set of the code $\mathcal{C}$.*

It is clear how to get the cardinality of $\mathcal{I}(\mathcal{C})$ when the code is over a field, however in the general case, things are more complicated as we shall see later.

We now introduce formally the systematic form of code. We first state the result for the easier case of codes over finite fields, and then give the tool necessary to present the general result over rings.

Proposition 5 (Systematic Form). *Let $\mathcal{C}$ be a systematic $[n,k]_q$-code. Any generator of this code has the form $[\mathbf{I_k} \mid \mathbf{A}]$, where $\mathbf{A} \in \mathbb{F}_q^{k\times(n-k)}$. Given a generator matrix of this form, a parity check matrix of the code is $[-\mathbf{A}^\top \mid \mathbf{I_{n-k}}]$. Note that $\mathcal{I}(\mathcal{C}) = [\![k]\!]$ and $\mathcal{I}(\mathcal{C}^\perp) = [\![n-k]\!] + k$.*

Before introducing the notion of systematic form for a code over a ring, we introduce some results about module theory. The following concepts are well-explained in [8], and we rephrase them for the sake of completeness. More general results about *chain rings* are presented in [18].

Definition 13 (p^m-Type of a Code). *We call $\log_{p^m}(h)$ the p^m-type of a code of type h.*

An important result from module theory states that any submodule of $(\mathbb{Z}/p^m\mathbb{Z})^n$, and thus any code, has a specific form.

Theorem 5 (Form of the Submodules [8]). *Given $\mathcal{C} \leq (\mathbb{Z}/p^m\mathbb{Z})^n$, then*

$$\mathcal{C} \cong \bigoplus_{i=1}^{m} (\mathbb{Z}/p^{m-i+1}\mathbb{Z})^{k_i}.$$

for some sequence $(k_1, \ldots, k_m)$. The latter is called the subtype of the code and it must respect the following constraint

$$\sum_{i=1}^{m} \frac{m-i+1}{m} k_i = k,$$

where k is the p^m-type of $\mathcal{C}$.

We relate these values to the size of the information set of the code.

Proposition 6 (Cardinality of the Information Set [19]). *Let $\mathcal{C}$ be a code over $\mathbb{Z}/p^m\mathbb{Z}$ of subtype $(k_1, \ldots, k_m)$ and $K = \sum_{i=1}^{m} k_i$. We have that $|\mathcal{I}(\mathcal{C})| = K$.*

We now have all the tools to introduce the systematic form of the generator and parity check matrix of any $\mathbb{Z}/p^m\mathbb{Z}$-code.

Proposition 7 (Generalized Systematic Form [19]). *Let $\mathcal{C} \leq (\mathbb{Z}/p^m\mathbb{Z})^n$ be a systematic code. Its generator matrix is*

$$\mathbf{G} = \begin{pmatrix} \mathbf{I_{k_1}} & \mathbf{A_{1,2}} & \mathbf{A_{1,3}} & \cdots & \mathbf{A_{1,m}} & \mathbf{A_{1,m+1}} \\ \mathbf{0} & p\mathbf{I_{k_2}} & \mathbf{pA_{2,3}} & \cdots & p\mathbf{A_{2,m}} & p\mathbf{A_{2,m+1}} \\ \mathbf{0} & \mathbf{0} & p^2\mathbf{I_{k_3}} & \cdots & p^2\mathbf{A_{3,m}} & p^2\mathbf{A_{3,m+1}} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & p^{m-1}\mathbf{I_{k_m}} & p^{m-1}\mathbf{A_{m,m+1}} \end{pmatrix},$$

where $\mathbf{A_{i,j}} \in (\mathbb{Z}/p^{m+1-i}\mathbb{Z})^{k_i \times k_j}$ *for* $j \leq m$ *and* $\mathbf{A_{i,m+1}} \in (\mathbb{Z}/p^{m+1-i}\mathbb{Z})^{k_i \times (n-K)}$. *The parity check matrix of* $\mathcal{C}$ *is*

$$\mathbf{H} = \begin{pmatrix} \mathbf{B_{1,1}} & \mathbf{B_{1,2}} & \cdots & \mathbf{B_{1,m-1}} & \mathbf{B_{1,m}} & \mathbf{I_{n-K}} \\ p\mathbf{B_{2,1}} & p\mathbf{B_{2,2}} & \cdots & p\mathbf{B_{2,m-1}} & p\mathbf{I_{k_m}} & \mathbf{0} \\ p^2\mathbf{B_{3,1}} & p^2\mathbf{B_{3,2}} & \cdots & p^2\mathbf{I_{k_{m-1}}} & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ p^{m-1}\mathbf{B_{m,1}} & p^{m-1}\mathbf{I_{k_2}} & \cdots & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix},$$

where $\mathbf{B_{i,j}} \in (\mathbb{Z}/p^{m+1-i}\mathbb{Z})^{k_{m-i+2} \times k_j}$ *for all* $i > 0$ *and* $\mathbf{B_{1,j}} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K) \times k_j}$.

Remark 4. If we consider a code over the finite field $\mathbb{Z}/p\mathbb{Z}$, then we have $m = 1$. By applying the proposition for the systematic form over rings, we retrieve nicely the systematic form stated for the field case in Proposition 5.

The fact that the parity check matrix generates $\mathcal{C}^\perp$ can be generalized to the ring case. Moreover, we have the following result about the dual:

Proposition 8 (p^m-Type of the Dual [8, Proposition 4]). *The dual code* $\mathcal{C}^\perp \leq (\mathbb{Z}/p^m\mathbb{Z})^n$ *has* p^m*-type* $(n - k)$.

Now that we have all the necessary tools from algebra and coding theory, we introduce the concept of a *PKE scheme*. This cryptographic scheme is the one that we will work with throughout this thesis.

2.4 Public-key Cryptography

Prior to the introduction of *Public-Key Cryptosystems (PKCs)*, security relied entirely on a shared and secret key between the two participants. This is also known as *Private-Key Cryptography*. The latter suffers from a limitation: both participants should have agreed on the shared key prior to the communication. In contrast, the public key approach allows both participants to communicate without any preliminary transmission. PKCs can be divided into two categories:

1. Digital Signature Algorithm (DSA);
2. Public-Key Encryption (PKE).

A Digital Signature Algorithm (DSA) is a system that ensures the authenticity and integrity of a message. In a PKE scheme, a message is encrypted with the intended recipient's public key and sent. Once received, it can be decrypted using the private key. The model is shown in Figure 2. The symbols pt, c, pk, sk, and $\mathcal{A}$ denote the plaintext, the ciphertext, the public key, the secret key, and the adversary respectively. The algorithms Enc, Dec and Gen are respectively the encryption, decryption, and generation algorithms.

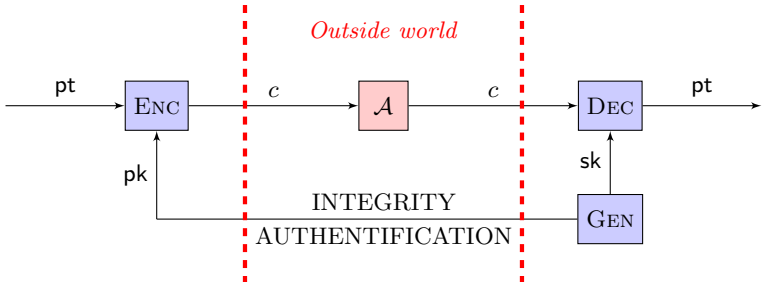


Fig. 2: Public-Key Encryption.

2.4.1 Post Quantum Public Key Cryptography Current public key cryptosystems rely on three mathematical problems:

1. Integer factorization;
2. Discrete logarithm;
3. Elliptic-curve discrete logarithm.

They can all be solved in quantum polynomial time using Shor's algorithm [41], which makes these cryptosystems insecure. Although the power required is far greater than what we can achieve today, we may have information that we want to protect for many years. In addition, we should also take into consideration the time that it will take to transform our tools into PQ ones [29]. There are two types of solutions for designing quantum-safe cryptosystems:

1. Quantum cryptography;
2. Post-Quantum (PQ) cryptography.

The former requires a quantum channel and therefore is not applicable in medium to long-range communication. The latter studies some classical cryptosystems which are based on a family of mathematical problems that are believed to be unbreakable even with a quantum computer. This family of problems can be divided into different categories such as hash-based, lattice-based, and code-based cryptography [7].

In 2016, the NIST issued a call for proposals for a PQ cryptography scheme, with a submission deadline at the end of 2017. Of the 23 DSA and 59 PKE/Key Encapsulation Mechanism (KEM) schemes, only seven were selected as finalists [1]. The length of their public key is shown in Table 1, from the lowest security level (Low) to the highest one. The "Type" column corresponds to one of the six categories presented above, and the "Security" column represents the security model achieved by the PKC. It's clear from this Table that the main drawback of the McEliece cryptosystem is the size of its public key.

As we will see in section 2.4.7, there is some hope that the public key size can be reduced using a variant of McEliece known as *Lee-McEliece*. This replaces the Hamming metric with the Lee metric. However, we will show in section 5 that this is not necessarily true over all rings.

Proposal	Type	$\|\mathsf{pk}\|$ Low (kB)	$\|\mathsf{pk}\|$ High (kB)	Security
PKE/KEM				
Classic McEliece	Code	261,120	1,357,824	IND-CCA2
CRYSTALS-KYBER	Lattice	1.632	3.168	IND-CCA
NTRU	Lattice	0.931	1.230	IND-CCA
SABER	Lattice	0.672	1.312	IND-CCA
DSA				
CRYSTALS-DILITHIUM	Lattice	1.312	2.592	SUF-CMA
FALCON	Lattice	0.897	1.793	EUF-CMA
Rainbow	Multivariate	157.8	1885.4	EUF-CMA

Table 1: Public key of the NIST finalists [1].

Throughout this thesis, we will focus on code-based cryptography and, especially, the McEliece cryptosystem and its variant, the Lee-McEliece cryptosystem. It has been shown that some code-based cryptosystems are not vulnerable to strong Fourier sampling, which is the method that is used in almost all exponential speed-up done by quantum algorithms [11]. Therefore, these kinds of cryptosystems are good candidates for PQ cryptosystems.

2.4.2 Code-based Cryptography The idea behind code-based cryptography is to use an error-correcting code as a trap door function. They are two ways to do this. The first is to add an error to a codeword, resulting in the *McEliece PKE* [28], and the second is to compute a syndrome relative to a parity check matrix, resulting in the *Niederreiter PKE* [33]. We can show that these two cryptosystems are equivalent [26]. The McElice cryptosystem is still secure today with some parameter adjustments. It is based on the decoding problem, which is difficult on average. We will come back to this problem and its hardness in more detail in Section 2.5.2.

2.4.3 Formalism We present the definition of a PKC, which is based on [21, Definition 11.1].

Definition 14 (PKC). *A Public-Key Cryptosystem is a quadruple*

$$(\textsc{Gen}, \textsc{Enc}, \textsc{Dec}, \mathcal{M})$$

such that:

- Gen *is a polynomial time probabilistic algorithm that takes as input the* ***security parameter*** 1^λ *and returns the pair of* ***keys*** $(\mathsf{pk}, \mathsf{sk})$.
- Enc *is a polynomial time probabilistic algorithm that takes as input the* ***public key*** pk *and a* ***plaintext*** pt *from the* ***message space*** $\mathcal{M}(\mathsf{pk})$. *It returns a* ***ciphertext****, denoted* c.

- DEC *is a polynomial time deterministic algorithm that takes as the* ***secret key*** sk *and a* ***ciphertext*** c *and returns a* ***plaintext*** pt *in the* ***message space*** $\mathcal{M}(\mathsf{pk})$ *or the failure symbol* $\perp$.
- $\mathcal{M}$ *maps a* ***public key*** pk *to a* ***message space*** $\mathcal{M}(\mathsf{pk})$.

Furthermore, we require the ***correctness*** *property:*

$$\forall r_g \quad \forall \mathsf{pt} \in \mathcal{M}(\mathsf{pk}) \quad \Pr_{r_e}[\textsc{Dec}(\mathsf{sk}, \textsc{Enc}(\mathsf{pk}, \mathsf{pt}; r_e) = \mathsf{pt}] = 1, \tag{3}$$

where $(\mathsf{pk}, \mathsf{sk}) \leftarrow \textsc{Gen}(1^\lambda; r_g)$

2.4.4 The McEliece Cryptosystem Codes are usually used in channel coding as a way of detecting and correcting errors. As we will see in Section 2.5.2, the problem of decoding a general code is NP-complete. We also believe that quantum computer does not have enough power to break this problem [11] and thus this problem is a good choice for a PQ PKC. We first introduce some notation

Notation 3 *We let* P_n *denote the set of permutation matrices of size* n *and* $\mathrm{GL}_n(\mathcal{R})$*, for a ring* $\mathcal{R}$*, the set of all non-singular matrices of size* n *over* $\mathcal{R}$.

Let n, k, t, and q be some integers and $\mathcal{F}$ be a family of efficiently decodable $[n, k, t]$-codes. We now describe the general McEliece scheme over $\mathbb{F}_q$ endowed with the Hamming weight.

Definition 15 (McEliece Cryptosystem). *A McEliece cryptosystem with parameter* $(n, k, t, q, \mathcal{F})$*, is a PKE scheme, denoted by* $\textsc{McEliece}^{\mathcal{F}}_{n,k,t,q}$*, defined by* $(\textsc{Gen}, \textsc{Enc}, \textsc{Dec}, \mathcal{M})$*, where*

$\textsc{Gen}(1^\lambda)$

1: $\mathcal{C} \leftarrow\!\!\$\ \mathcal{F}$
2: $\mathbf{G} \leftarrow \textsc{GeneratorMatrixOf}(\mathcal{C})$
3: $\mathbf{P} \leftarrow\!\!\$\ \mathrm{P}_n$
4: $\mathbf{S} \leftarrow\!\!\$\ \mathrm{GL}_k(\mathbb{F}_q)$
5: $\hat{\mathbf{G}} \leftarrow \mathbf{SGP}$
6: **return** $(\mathsf{pk} = (\hat{\mathbf{G}}, t), \mathsf{sk} = (\mathbf{S}, \mathbf{G}, \mathbf{P}))$

$\textsc{Enc}(\mathsf{pk} = (\hat{\mathbf{G}}, t), \mathbf{x})$

1: $\mathbf{e} \leftarrow\!\!\$\ \{\mathbf{e} \in \mathbb{F}_q^n \mid wt_H(\mathbf{e}) = t\}$
2: **return** $\mathbf{x}\hat{\mathbf{G}} + \mathbf{e}$

, are the respective generation and encryption algorithms,

$\textsc{Dec}(\mathsf{sk} = (\mathbf{S}, \mathbf{G}, \mathbf{P}), \mathbf{y})$

1: $\hat{\mathbf{y}} \leftarrow \mathbf{y}\mathbf{P}^{-1}$
2: $\hat{\mathbf{m}} \leftarrow \textsc{EfficientDecoding}_{\mathcal{C}}(\hat{\mathbf{y}})$
3: **return** $\hat{\mathbf{m}}\mathbf{S}^{-1}$

is the decryption algorithm, EFFICIENTDECODING *is a deterministic and efficient algorithm that given any* $\mathbf{y} \in B^n_{q,H}(\mathbf{x}, t)$ *returns* t *for any* $\mathbf{x} \in \mathcal{C} \in \mathcal{F}$ *and* $\mathcal{M} \equiv \mathbb{F}_q^k$.

Theorem 6 (Correctness of McEliece Cryptosystem). *For any family $\mathcal{F}$ of efficiently decodable $[n,k,t]$-code over $\mathbf{F}_q$, the cryptosystem* $\textsc{McEliece}^{\mathcal{F}}_{n,k,t,q}$ *respects the correctness property.*

Proof. We can specify the randomness of Gen with $\mathbf{G}$ a generator matrix of a code $\mathcal{C} \in \mathcal{F}$, $\mathbf{P} \in \mathrm{P}_n$ and $\mathbf{S} \in \mathrm{GL}_k(\mathbb{F}_q)$ and the randomness of Enc with $\mathbf{e} \in \{\mathbf{e} \in \mathbb{F}_q^n \mid \mathrm{wt}_H(\mathbf{e}) = t\} = S^n_{q,H}(\mathbf{0}, t)$. We fix $\mathbf{G}$, $\mathbf{P}$ and $\mathbf{S}$ arbitrary, and get $((\hat{\mathbf{G}}, t), (\mathbf{S}, \mathbf{G}, \mathbf{P})) \leftarrow \textsc{Enc}(1^\lambda; \mathbf{G}, \mathbf{P}, \mathbf{S})$. Finally, we fix an arbitrary plaintext $\mathbf{x} \in \mathbb{F}_q^k$. Note that by definition, both $\mathbf{P}$ and $\mathbf{S}$ are invertible and that $\mathrm{wt}_H(\mathbf{e}\mathbf{P}^{-1}) = t$.

$$
\begin{aligned}
&\Pr_{\mathbf{e} \in S^n_{q,H}(\mathbf{0},t)}[\textsc{Dec}(\mathbf{S}, \mathbf{G}, \mathbf{P}, \textsc{Enc}(\hat{\mathbf{G}}, t, \mathbf{x})) = \mathbf{x}] \\
&\quad = \Pr_{\mathbf{e} \in S^n_{q,H}(\mathbf{0},t)}[\textsc{Dec}(\mathbf{S}, \mathbf{G}, \mathbf{P}, \mathbf{x}\hat{\mathbf{G}} + \mathbf{e}) = \mathbf{x}] \\
&\quad = \Pr_{\mathbf{e} \in S^n_{q,H}(\mathbf{0},t)}[\textsc{EfficientDecoding}((\mathbf{x}\hat{\mathbf{G}} + \mathbf{e})\mathbf{P}^{-1})\mathbf{S}^{-1} = \mathbf{x}] \\
&\quad = \Pr_{\mathbf{e} \in S^n_{q,H}(\mathbf{0},t)}[\textsc{EfficientDecoding}((\mathbf{x}\mathbf{S}\mathbf{G}\mathbf{P} + \mathbf{e})\mathbf{P}^{-1})\mathbf{S}^{-1} = \mathbf{x}] \\
&\quad = \Pr_{\mathbf{e} \in S^n_{q,H}(\mathbf{0},t)}[\textsc{EfficientDecoding}((\mathbf{x}\mathbf{S}\mathbf{G} + \mathbf{e}\mathbf{P}^{-1})\mathbf{S}^{-1} = \mathbf{x}] \\
&\quad = \Pr_{\mathbf{e} \in S^n_{q,H}(\mathbf{0},t)}[\mathbf{x}\mathbf{S}\mathbf{S}^{-1} = \mathbf{x}] = \Pr_{\mathbf{e} \in S^n_{q,H}(\mathbf{0},t)}[\mathbf{x} = \mathbf{x}] = 1.
\end{aligned}
$$

This shows that the encryption system is correct. □

In his first proposal, McEliece proposed *binary Goppa codes* as a family of efficiently decodable codes [28]. We will not go into the details of these codes since we are interested in decoding random linear codes and not specific ones. In the next Section, we argue why this cryptosystem is considered to be secure.

2.4.5 Security The security of the McEliece PKC relies on two computational assumptions [35].

Assumption 1 (Indistinguishability) *The matrix $\hat{\mathbf{G}}$ generated by* Gen *is computationally indistinguishable from a uniformly random matrix of the same size.*

Remark 5. It is not true that there is no distinguisher that can differentiate a Goppa code from a random binary code since such adversaries exist when the rate of the code is high [12].

Assumption 2 (Hardness of Decoding) *The problem of decoding general linear code is, on average computationally hard, even with a quantum computer.*

Without going into detail, the first assumption guarantees that no information about the chosen secret code leaks into the public key, while the second assumption guarantees that the ciphertext cannot be decoded without knowing

the secret key. The second assumption is discussed in Section 2.5.2. As depicted in Table 1, McEliece Cryptosystem can be transformed, so it becomes IND-CCA2-secure [35], which is equivalent to semantic security. The next Section explains how one can set the size of the public key in order to achieve a desired security goal.

2.4.6 Choosing the Public Key Size We fix the size of the public key by looking at the complexity of the best-known algorithm that can break the cryptosystem. In the case of the McEliece cryptosystem, there are two types of decoding algorithms:

1. Statistical Decoding.
2. Information Set Decoding (ISD);

The latter performs better than the former and thus, all of public key size analyses are based on the complexity of ISD algorithms. We will introduce Statistical Decoding in Section 3 and ISD in Section 4.

An analysis of the public key size was performed in [32] using the Lenstra-Verheul's model [24]. They state its size must be at least 131 kB in order to secure data until 2050. This is huge and is the main issue with the McEliece cryptosystem. In the next section, we will introduce a generalization of the McEliece cryptosystem over a prime power integer residue ring embedded with the Lee metric. This cryptosystem is called the *Lee-McEliece* PKC. Using the generalization of the ISD algorithms to the Lee metric [19], it has been shown that the size of the public key could be reduced to 53 kB in order to achieve 128-bit security [22]. However, the proof does not take into consideration how statistical decoding might be generalized to the Lee metric.

This new algorithm is a natural transformation of the classical ISD and it does not take into account the structure of the Lee metric to improve its performance. Another type of ISD attack has been proposed by Jessica Barfiffi *et al.* which takes advantage of the Lee metric through restricted balls [3].

We will show later that the statistical decoding algorithm and its generalization, the reduction to the Learning Parity with Noise (LPN) technique, are no less affected than the ISD techniques when changing the metric and the base field. This raises the following question: Is there a point, at which the statistical methods outperform ISD ones? We will answer this question in Section 5. The next section discusses the Lee-McEliece cryptosystem.

2.4.7 Lee-McEliece Cryptosystem We give the formal definition of the Lee-McEliece cryptosystem over $\mathbb{Z}/p^m\mathbb{Z}$.

Definition 16 (Lee-McEliece Cryptosystem). *Let $\mathcal{F}$ be any family of efficiently decodable codes over $\mathbb{Z}/p^m\mathbb{Z}$ in the Lee metric, of subtype $\mathbf{k} = (k_1, \ldots, k_m)$, that can correct t errors. A Lee-McEliece Cryptosystem with parameter*

$$(p, m, n, \mathbf{k}, t, q, \mathcal{F}),$$

is a PKE scheme, denoted by $\text{LEE-MCELIECE}^{\mathcal{F}}_{p,m,n,\mathbf{k},t,q}$*, defined by*

$$(\text{GEN}, \text{ENC}, \text{DEC}, \mathcal{M}), \tag{4}$$

where

$\text{GEN}(1^\lambda)$

1: $\mathcal{C} \leftarrow\!\!\$\ \mathcal{F}$
2: $\mathbf{G} \leftarrow \text{GENERATORMATRIXOF}(\mathcal{C})$
3: $\mathbf{P} \leftarrow\!\!\$\ \mathrm{P}_n$
4: $\mathbf{S_1} \leftarrow\!\!\$\ \mathrm{GL}_{k_1}(\mathbb{Z}/p^m\mathbb{Z})$
$\vdots$
$\mathbf{S_m} \leftarrow\!\!\$\ \mathrm{GL}_{k_m}(\mathbb{Z}/p^m\mathbb{Z})$
5: $\mathbf{S} \leftarrow \begin{pmatrix} \mathbf{S_1} & \mathbf{0} & \mathbf{0} \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S_2} & \mathbf{0} \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \cdots & \mathbf{0} & \mathbf{S_m} \end{pmatrix}$
6: $\hat{\mathbf{G}} \leftarrow \mathbf{SGP}$
7: **return** $(\mathsf{pk} = (\hat{\mathbf{G}}, t), \mathsf{sk} = (\mathbf{S}, \mathbf{G}, \mathbf{P}))$

$\text{ENC}(\mathsf{pk} = (\hat{\mathbf{G}}, t), \mathbf{x})$

1: $\mathbf{e} \leftarrow\!\!\$\ \{\mathbf{e} \in (\mathbb{Z}/p^m\mathbb{Z})^n \mid wt_L(\mathbf{e}) = t\}$
2: **return** $\mathbf{x}\hat{\mathbf{G}} + \mathbf{e}$

are the respective generation and encryption algorithms,

$\text{DEC}(\mathsf{sk} = (\mathbf{S}, \mathbf{G}, \mathbf{P}), \mathbf{y})$

1: $\hat{\mathbf{y}} \leftarrow \mathbf{y}\mathbf{P}^{-1}$
2: $\hat{\mathbf{m}} \leftarrow \text{EFFICIENTDECODING}_{\mathcal{C}}(\hat{\mathbf{y}})$
3: **return** $\hat{\mathbf{m}}\mathbf{S}^{-1}$

is the decryption algorithm, EFFICIENTDECODING *is a deterministic and efficient decoding algorithm for any* $\mathcal{C} \in (\mathbb{Z}/p^m\mathbb{Z})^K$ *and* $\mathcal{M} \equiv (\mathbb{Z}/p^m\mathbb{Z})^K$ *for* $K = |\mathcal{I}(\mathcal{C})| = \sum_{i=0}^{m} k_i$ *by Proposistion 6.*

The proof of correctness is similar to the one of the McEliece cryptosystem.

Theorem 7 (Correctness of Lee-McEliece Cryptosystem). *For any family* $\mathcal{F}$ *of efficiently decodable codes over* $\mathbb{Z}/p^m\mathbb{Z}$ *in the Lee metric, of subtype* $\mathbf{k} = (k_1, \ldots, k_m)$*, that can correct* t *errors.* $\text{LEE-MCELIECE}^{\mathcal{F}}_{p,m,n,\mathbf{k},t,q}$ *respects the correctness property.*

2.5 Coding Theory

The McEleice PKE and its variant, the Lee-McEliece PKE are based on error-correcting codes. The main goal of error-correcting codes is to add useful redundancy to the data from a source in order to make the transmission, over a noisy channel, more robust. The original use of error-correcting codes is depicted in

Figure 3. The encoded plaintext is denoted by y and by y' after being affected by the noise. The indistinguishability assumption (Assumption 1) combined with the fact that only some specific codes are efficiently decodable (Assumption 2), makes error-correcting codes into a good candidate for a trapdoor function.

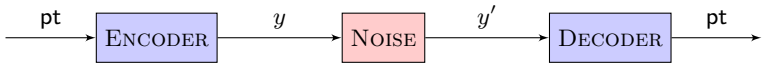


Fig. 3: Error-correcting code.

2.5.1 The Decoding Problems We introduce the formal definition of the Syndrome Decoding Problem (SDP) and its variant, the exact SDP.

Problem 1 (Syndrome Decoding Problem). Given a parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, a syndrome $\mathbf{s} \in \mathbb{F}_q^{n-k}$ of a code, an integer t and a weight function wt, find an error $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{He}^\top = \mathbf{s}^\top$ and $\mathrm{wt}(\mathbf{e}) \leq t$.

In particular, we will be interested in Hamming and Lee's version of these problems, which we will denote by *Hamming-SD* and *Lee-SD*. In these variants, the inequality $\mathrm{wt}(\mathbf{e})$ can be equivalently changed to $\mathbf{e}$ lying respectively in $B_{q,H}^n(\mathbf{x}, t)$ and $B_{q,L}^n(\mathbf{x}, t)$. We can restrict the problem above by asking to return an error that has weight exactly t.

Problem 2 (Exact Syndrome Decoding Problem). Given a parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, a syndrome $\mathbf{s} \in \mathbb{F}_q^{n-k}$ of a code, an integer t and a weight function wt, find an error $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{He}^\top = \mathbf{s}^\top$ and $\mathrm{wt}(e) = t$.

As before, we consider the Hamming and Lee versions of this problem, denoted by *Hamming-ESD* and *Lee-ESD*. Obviously, if you can solve these problems, you can decrypt and solve the decryption problem. However, the reverse is probably not true. The assumption 1 is not true for any family of codes, and so in these cases, the decryption problem is restricted to a subset of all possible codes.

2.5.2 The Hardness of Decoding In the Hamming metric, E. Berlekamp, R. McEliece, and H. van Tilborg proved in 1978 that the decoding problems are NP-complete for random binary linear codes [5]. Then, Michael Alekhnovich showed that it is hard to decode binary random code on average, thus proving that there exists a secure cryptosystem based on the intractability of decoding random binary linear codes. The intractability has been generalized in the Lee metric by Violetta Weger *et al.* in [47]. We summarize these results in the following theorem:

Theorem 8 (Complexity of the Decoding Problems). *The Hamming-SD, Hamming-ESD, Lee-SD, and Lee-ESD problems are* NP*-complete.*

Note that the average hardness of these problems does not imply that they cannot be solved in polynomial time by a quantum computer. Therefore the NP $\neq$ P assumption is not enough to motivate Assumption 2. The motivation behind this assumption is that almost all exponential speed-up made by quantum computers use the method of *strong Fourier sampling* and that the decoding problems are immune to this method [11].

Some linear codes are easily decodable, otherwise, channel coding would be of no interest. Here are some families of codes with known efficient decoding algorithms:

- Hamming codes [16];
- Hadamard codes;
- Simplex codes;
- Goppa codes [13].

If we restrict the Goppa codes to the binary field, we get the binary Goppa codes. That is the family of codes used in the McEliece PKE. We will not study these codes since we are interested in general decoding.

3 Statistical Decoding

Before introducing the state-of-the-art *reduction to LPN* technique in Section 5, we must first understand classical statistical decoding algorithms. The idea of statistical decoding was first introduced in 2001 by Al Jabri [20]. The first use case was decoding linear code on the Hamming metric, and for this purpose, the algorithm was less efficient than the ISD ones. We first explain how statistical decoding can be generalized to rings and how to translate it into the Lee metric. We show that the change of metric does not increase the running time of the algorithm, which is not the case when considering ISD algorithm [10]. The two types of algorithms are compared in Section 5.

3.1 The Statistical Bias

In statistical decoding, we are given a $[n, k, d]$-code $\mathcal{C}$, and want to solve the Hamming-SDP given an instance $(\mathbf{H}, \mathbf{s}, t)$. To do so, we compute a set of parity check equations $\mathcal{H}_w \subseteq \mathcal{C}^\perp$, where each element of $\mathcal{H}_w$ has a Hamming weight w and $w < n/2$. Without loss of generality, we could also consider the case $w > n/2$.

To explain the idea behind this algorithm, we first restrict ourselves to the binary world. Consider the following equation,

$$\forall \mathbf{h} \in \mathcal{H}_w \quad \mathbf{y}\mathbf{h}^\top = \mathbf{e}\mathbf{h}^\top = b \in \mathbb{Z}/2\mathbb{Z},$$

where the equality comes from the fact that $\mathbf{h}$ is an element of the dual code $\mathcal{C}^\perp$. We note that the value b gives some information about the error vector $\mathbf{e}$. Indeed, if $b = 1$, the vectors $\mathbf{y}$ and $\mathbf{h}$ must have an odd number of indices in which both

vectors have components equal to 1. Say differently, $|\text{Supp}(\mathbf{y}) \cap \text{Supp}(\mathbf{h})|$ must be odd. In that case, we say that $\mathbf{h}$ provides an *odd error detection*. Conversely, if $b = 0$, $\mathbf{h}$ is said to have provided an *even error detection*. Note that in the latter case, it is possible that no error has been detected.

In the case $b = 1$, there must be at least one error bit of $\mathbf{e}$ that corresponds to a non-zero bit of $\mathbf{h}$. Thus, in a sense, $\mathbf{h}$ is a way to vote for the error bits. A first approach to solve the decoding problem would be to sum all the vectors $\mathbf{h}$ in $\mathcal{H}^{\perp}$ that provides odd error detection. We would end up with a vector $\mathbf{v} \in \mathbb{N}^n$ that is biased on the erroneous indices. Therefore, by considering the indices of the largest t values in $\mathbf{v}$, we have a chance to find the indices of the non-zeroes entries of $\mathbf{e}$ and thus $\mathbf{e}$. Finding $\mathbf{e}$ lets us solve the decoding problem. This algorithm is presented in the next Section. To demonstrate this point, we propose the following example.

Example 2 (The statistical bias). We let $\mathbf{e} = [1, 1, 0, 0, 0, 0, 0] \in \mathbb{F}_2^7$ be the error vector and $w = 3$. We are interested in testing different cardinalities of $\mathcal{H}_w$ and comparing the difference in $\mathbf{v} = \sum_{\mathbf{h} \in \mathcal{H}_w} (\mathbf{y}\mathbf{h}^\top)\mathbf{h}$. We normalized the value of $\mathbf{v}$ into relative frequencies by dividing each of its components by the L_1-norm of $\mathbf{v}$, that is, by the sum of all its components over $\mathbb{Z}$.

Figure 4 depicts the relative frequencies of all bits under the different sizes of $\mathcal{H}_w$. We consider in this example the size: $2 \cdot 10^1$; $2 \cdot 10^2$; $2 \cdot 10^3$ and $2 \cdot 10^4$. Looking at the two largest $|\mathcal{H}_w|$ histograms (4c and 4d), it is clear that the two first bits of the error vector are those that are not zero. Analyzing Subfigure 4b, one could assume that $\mathbf{e} = [1, 0, 0, 0, 0, 1, 0]$, which is false. Moreover, the case with the smallest $\mathcal{H}_w$, looks relatively random. Thus, it seems that in our case, the number of parity check equations needed is at least 2000 which is large compared to the number of error bits and the vector size.

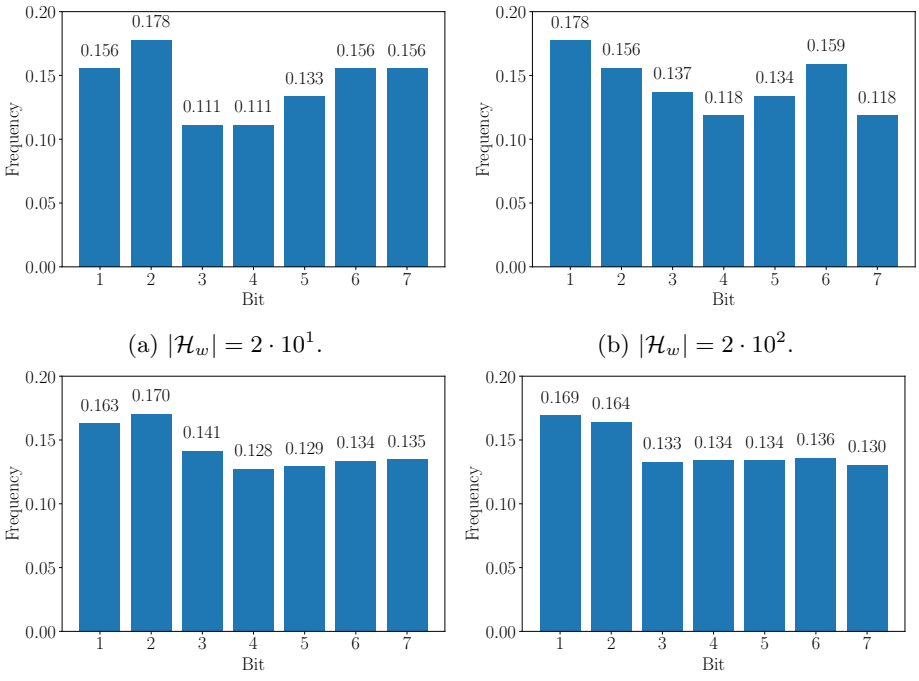
This example and our observations should raise the following questions:

1. How big does $\mathcal{H}_w$ have to be to solve the problem with high probability?
2. How to compute $\mathcal{H}_w$?
3. Is there a better way to retrieve information about $\mathbf{e}$ using $\mathcal{H}_w$?

The following Section is devoted to the first question. The second question will not be considered in this thesis. Moreover, $\mathcal{H}_w$ can be precomputed and therefore, the complexity of its computation can be negated in a cryptographic context. When comparing the time complexities in Section 5, we will always consider the two following cases, statistical decoding with its precomputation and statistical decoding without its precomputation. A solution to the last question is proposed in [34]. We will not address it in this thesis.

3.2 The Algorithm

From this point on, we will assume that the decoding algorithm has an additional parameter $\mathcal{H}_w$, which is the set of parity check equations, and that it has been calculated in advance.



(a) $|\mathcal{H}_w| = 2 \cdot 10^1$.

(b) $|\mathcal{H}_w| = 2 \cdot 10^2$.

(c) $|\mathcal{H}_w| = 2 \cdot 10^3$.

(d) $|\mathcal{H}_w| = 2 \cdot 10^4$.

Fig. 4: Frequencies of the counting vector for different sizes of $\mathcal{H}_w$ in the statistical decoding when $t = 2$ and $w = 3$ and $\mathbf{e} = [1, 1, 0, 0, 0, 0, 0]$ over $\mathbb{F}_2^7$.

3.2.1 The Binary Case We present the statistical decoding algorithm. It was first presented by Al Jabri in 2001 [20].

Binary Statistical Decoding($\mathcal{H}_w$, $\mathbf{y}$)

1 : // Where the outer sum is done over $\mathbb{Z}$

2 : // whereas the inner and scalar product are done over $\mathbb{F}_2$

3 : $\mathbf{v} \leftarrow \sum_{\mathbf{h} \in \mathcal{H}_w} (\mathbf{y}\mathbf{h}^\top)\mathbf{h}$

4 : // Get the set of non-erroneous indices

5 : $I \leftarrow \{\text{Indices of the } k \text{ smallest values of } \mathbf{v} \text{ such that } \mathbf{G}^{\mathfrak{C}(I)} \text{ is invertible}\}$

6 : // Invert the codeword on its information set

7 : **return** $\mathbf{y}_I(\mathbf{G}^{\mathfrak{C}(I)})^{-1}$

Assuming that $\mathcal{H}_w$ is precomputed, we can give the running time with respect to the cardinality of $\mathcal{H}_w$.

Theorem 9 (Complexity of Statistical Decoding). *Suppose that $\mathcal{H}_w$ is precomputed. Then the statistical decoding needs*

$$T_{\text{SD}}(|\mathcal{H}_w|) = 2n|\mathcal{H}_w| + nk + k^3 + k^2$$

binary operations.

Proof. At every summation step, we must first calculate the inner product, which takes n steps and finally, if the result is not zero, add to $\mathbf{v}$, which also takes n steps. The naive way to take the k smallest elements within a set of n ones uses nk binary operations. Finally, using the Gauss-Jordan elimination and the schoolbook multiplication, the last line needs $k^3 + k^2$ operations. □

Remark 6. By Theorem 8, we know that the problem of computing $\mathcal{H}_w$ must be NP-hard. To see suppose for the sake of contradiction that it is not NP-hard. This would imply that $|\mathcal{H}_w|$ is of polynomial size and thus by the closure property of polynomials, T_{SD} would also be polynomial in the input size. Finally, we would get that the sum of the precomputation and the computation itself is polynomial, which is a contradiction.

Observe that in the proof, we only consider the most naive algorithms. We could improve the algorithm by using a better algorithm for matrix inversion, matrix/vector multiplication, and finding the k smallest element in a set of n elements.

The success probability of this algorithm depends on the cardinality of $\mathcal{H}_w$, as shown in Example 2. Section 3.3 presents an expression for the cardinality to ensure that the algorithm runs successfully with high probability. We now extend the algorithm over any finite field.

3.2.2 Over $\mathbb{F}_q$

The extension of the algorithm over $\mathbb{F}_q$ is done naturally. It was first presented by Niebuhr in 2011 [31]. The difference with the binary field concern mainly the analysis of the cardinality of $\mathcal{H}_w$, the algorithm itself is very similar. The analysis of the cardinality is done in Section 3.3. Since over $\mathbb{F}_q$ non-zero elements are not necessarily 1, we must introduce the following function to count correctly while summing over all elements of $\mathcal{H}_w$.

Definition 17. *Let*

$$\begin{aligned} \pi \colon \mathbb{F}_q &\to \{0,1\} \\ x &\mapsto \mathbb{1}_{x\neq 0}, \end{aligned}$$

and $\Pi \colon \{0,1\}^n \to \{0,1\}^n$ its component-wise extension.

Here is the pseudo-code of the general statistical decoding algorithm.

Finite Field Statistical Decoding($\mathcal{H}_w$, $\mathbf{y}$)

1 : // Where the outer sum and the scalar product are done over $\mathbb{Z}$
2 : // whereas the inner product is over $\mathbb{F}_q$
3 : $\mathbf{v} \leftarrow \sum_{\mathbf{h} \in \mathcal{H}_w} \Pi((\mathbf{y}\mathbf{h}^\top)\mathbf{h})$
4 : // Get the set of non-erroneous indices
5 : $I \leftarrow$ {Indices of the k smallest values of $\mathbf{v}$ such that $\mathbf{G}^{\mathfrak{C}(I)}$ is invertible}
6 : // Invert the codeword on its information set
7 : **return** $\mathbf{y}_I(\mathbf{G}^{\mathfrak{C}(I)})^{-1}$

As in the binary case, the running time of this algorithm depends essentially on the cardinality of $\mathcal{H}_w$. Assuming that $\mathcal{H}_w$ is precomputed, we can give the running time with respect to the cardinality of $\mathcal{H}_w$.

Theorem 10 (Complexity of Statistical Decoding over Finite Field). *Suppose that $\mathcal{H}_w$ is precomputed and that field's operations take constant time. The statistical decoding over any finite field needs*

$$T'_{\text{SD}}(|\mathcal{H}_w|) = 2n^2|\mathcal{H}_w| + nk + k^3 + k^2$$

binary operations.

Proof. The analysis is the same as in Theorem 9 expect that each iteration of the sum takes one more step, namely, computing $\Pi(\cdot)$. This last step takes n more steps. □

3.3 The Cardinality of $\mathcal{H}_w$

As we have seen, the number of parity checks equations is a dominant factor in the time complexity of statistical decoding, whether with or without precomputation. In addition, we want a sufficient quantity of these equations in order to recover the non-zeroes of $\mathbf{e}$ with enough probability. This Section is devoted to expressing some approximation on the size of $\mathcal{H}_w$ to have at least 95% chance of decoding $\mathbf{y}$.

3.3.1 The Binary Case Let $\mathcal{C}_w^\perp$ denote the subset of $\mathcal{C}^\perp$ of vectors of Hamming weight w. We specify the bit-wise versions of the notions of *odd and even error detection*. We say that the vector $\mathbf{h}$ has provided *odd/even error detection in bit i* if $\mathbf{h}_i = 1$ and $\mathbf{h}\mathbf{y}^\top$ is respectively 1 and 0. Knowing that $\mathbf{h}\mathbf{e}^\top = 1$, $\mathbf{h}_i$ either follows a Bernoulli distribution of parameter $p^+_{w,t}$ or $q^+_{w,t}$. Our goal is to decide which of these two distributions $\mathbf{h}_i$ follows and, consequently, decide whether i is an erroneous position of the codeword. This leads to the definition of the two respective probabilities.

Definition 18. *Let $p^+_{w,t}$ and $q^+_{w,t}$ be the following probabilities:*

$$p^+_{w,t} = \Pr_{\mathbf{h}\in\mathcal{C}_w^\perp}[\mathbf{h}_i = 1 \mid \mathbf{e}_i = 1 \wedge \mathbf{hy}^\top = 1],$$
$$q^+_{w,t} \triangleq \Pr_{\mathbf{h}\in\mathcal{C}_w^\perp}[\mathbf{h}_i = 1 \mid \mathbf{e}_i = 0 \wedge \mathbf{hy}^\top = 1].$$

For all $i \in [\![n]\!]$, these probabilities are easily seen to be the following expressions:

Proposition 9 ([20]). *The probabilities $p^+_{w,t}$ and $q^+_{w,t}$ can be written as the following expressions:*

$$p^+_{w,t} = \frac{\sum_{m\ odd}^{m\le t}\binom{t-1}{m-1}\binom{n-t}{w-m}}{\sum_{m\ odd}^{m\le t}\binom{t}{m}\binom{n-t}{w-m}},$$
$$q^+_{w,t} = \frac{\sum_{m\ odd}^{m\le t}\binom{t}{m}\binom{n-t-1}{w-m-1}}{\sum_{m\ odd}^{m\le t}\binom{t}{m}\binom{n-t}{w-m}}.$$

Note that since $w < n/2$, the probability $p^+_{w,t}$ is strictly greater than $q^+_{w,t}$ and that they do not depend on $\mathbf{e}_j$, for $j \neq i$. One may ask if it is possible to use even error detection to learn something more about $\mathbf{e}$. This question has motivated Overbeck's improvement [34]. The latter makes use of the even error detections to learn more about $\mathbf{e}$. We will not discuss this improvement in this thesis.

A first proposal on the size of $\mathcal{H}_w$ was suggested by Al Jabri [20]. He states that $\mathcal{H}_w$ requires a cardinality of

$$625 \cdot 10^{-6} \cdot p^+_w \left(1 - p^+_w\right) \epsilon^{-2} \tag{5}$$

in order to ensure with 95% probability that erroneous indices of the relative frequencies $\mathbf{v}/|\mathcal{H}|$ are within ϵ of p.

However, the experiments in [34] proved that this value is far too optimistic by a factor of 2^{13}. Therefore, the author of [34] provides a more realistic approximation of $\mathcal{H}_w$ that coincides with the factor of 2^{13},

Proposition 10 (Cardinality of $\mathcal{H}_w$ over Binary Field [34]). *Over the binary field, in order for statistical decoding to achieve 95% probability of success we need $\mathcal{H}_w$ to be of the following size:*

$$|\mathcal{H}_w| \approx 5.4 p^+_w \frac{(1 - p^+_w)}{\left(p^+_w - q^+_w\right)^2}.$$

3.3.2 Over $\mathbb{F}_q$ Niebuhr [31] generalized the previous proposition to the general Galois field $\mathbb{F}_q$. Note that since the alphabet of the ambient space is no longer binary, we need to change our definition of odd error detection. Thus, we define an odd error detection in bit i if $\mathbf{h}_i \neq 0$ and $\mathbf{hy}^\top \neq 0$. We also define the following quantity,

Definition 19. *Let i be an integer and q a strictly positive integer, we let*

$$\mathfrak{A}(q,i) = \left\lfloor \frac{(q-1)^i}{q} \right\rceil.$$

The equation above represents the number of ways in which i elements of $\mathbb{F}_q$ can sum up to a specific non-zero element $\mathbb{F}_q$. In other words, this is equivalent to the number of ways we can choose two vectors of size i with entries in $[\![q-1]\!]$ such that their inner product yields a given non-zero constant. We can now adapt the probabilities $p^+_{w,t}$ and $q^+_{w,t}$ into the finite field case.

Definition 20. *Let t and w be some integers and q be a prime power, we define the two following quantities:*

$$p^{++}_{w,t,q} = \Pr_{\mathbf{h}\in\mathcal{C}^\perp_w}[\mathbf{h}_i \neq 0 \mid \mathbf{e}_i \neq 0 \wedge \mathbf{h}\mathbf{y}^\top \neq 0],$$
$$q^{++}_{w,t,q} = \Pr_{\mathbf{h}\in\mathcal{C}^\perp_w}[\mathbf{h}_i \neq 0 \mid \mathbf{e}_i = 0 \wedge \mathbf{h}\mathbf{y}^\top \neq 0].$$

These new probabilities can be expressed as follows. Note that there is probably a typo in [31, Equation 5] in the $\binom{n-t}{w-j}$ of the numerator. This coefficient should reflect the number of ways we can distribute $w-j$ remaining non-zero entries of $\mathbf{h}$ within the zero entries of $\mathbf{e}$. But knowing that $\mathbf{e}_i = 0$ and $\mathbf{h}_i \neq 0$ means that we can only distribute $w-j-1$ non-zero elements of $\mathbf{h}$ within $n-t-1$ zero elements of $\mathbf{e}$.

Proposition 11 ([31]). *These two probabilities can be expressed using $\mathfrak{A}$ as follows:*

$$p^{++}_{w,t,q} = \frac{\sum_{j=1}^{t} \mathfrak{A}(q,j)\binom{t-1}{j-1}\binom{n-t}{w-j}(q-1)^{w-j}}{\sum_{j=1}^{t} \mathfrak{A}(q,j)\binom{t}{j}\binom{n-t}{w-j}(q-1)^{w-j}},$$
$$q^{++}_{w,t,q} = \frac{\sum_{j=1}^{t} \mathfrak{A}(q,j)\binom{t}{j}\binom{n-t-1}{w-j-1}(q-1)^{w-j}}{\sum_{j=1}^{t} \mathfrak{A}(q,j)\binom{t}{j}\binom{n-t}{w-j}(q-1)^{w-j}}.$$

As in the binary case, these probabilities do not depend on $\mathbf{e}_j$, for $j \neq i$. Finally, we transform the estimation of the required cardinality of $\mathcal{H}_w$ so it also works over $\mathbb{F}_q$:

Proposition 12 (Cardinality of $\mathcal{H}_w$ over $\mathbb{F}_q$ [31]). *Over $\mathbb{F}_q$, in order for statistical decoding to achieve 95% probability of success we need $\mathcal{H}_w$ to be of the following size:*

$$|\mathcal{H}_w| \approx 2.72 \frac{q}{q-1} p^{++}_w \frac{1-p^{++}_w}{\left(p^{++}_w - q^{++}_w\right)^2}.$$

If the difference in probabilities is ignored, this approximation is $\frac{q}{q-1}$ larger than the binary one. The experiments also showed that the size of the field does not affect the required size of $\mathcal{H}_w$ [31]. This is a major difference with ISD since the

latter is badly impacted when using larger fields, as we will show in Section 4. In particular, for sufficiently large fields, the statistical decoding becomes more efficient than the algorithms of the ISD family. This is a crucial discovery since over the binary field, it has been proven that it is very unlikely that we can attack the McEliece cryptosystem with statistical decoding and that ISD algorithms remain the best ones [34].

This lead to the following questions: Is it the case in non-binary versions of this cryptosystem? And what about the Lee metric? The next section presents how we can extend the algorithm to finite rings and analyze the required cardinality of $\mathcal{H}_w$. Both of the above questions are answered in Section 5. In the latter, we compare ISD techniques to statistical decoding under various rings and the Hamming and Lee metrics.

3.4 Statistical Decoding over Rings

Compared to $\mathbb{F}_q$, the probability expressions for the statistical bias over rings are not as nice as those presented in the equations of Propositon 11. This is because the structure of a ring is not as good as the one of a field. For example, in a ring, we must consider zero divisors. However, the algorithm is the same as the one for $\mathbb{F}_q$ and so is its complexity with respect to $|\mathcal{H}_w|$. One could ask how to handle matrix inversion over rings. Gaussian elimination may not work since there are some non-zero elements that are not invertible. The following section explains how to invert matrices over $\mathbb{Z}/m\mathbb{Z}$ and argues why changing to a prime power residue field does affect the algorithm.

3.4.1 Inverting Matrices over $\mathbb{Z}/m\mathbb{Z}$ Like for matrices over fields, being able to apply *Gaussian eliminations* over finite rings is crucial in algebra. Moreover, both statistical decoding and ISD algorithms need this tool in order to invert matrices. However, this algorithm does not always works, even for invertible matrix as depicted in the next example.

Example 3. Let

$$\mathbf{M} = \begin{pmatrix} 2 & 2 & 3 \\ 0 & 1 & 3 \\ 1 & 0 & 1 \end{pmatrix} \in (\mathbb{Z}/4\mathbb{Z})^{3\times 3}.$$

Note that $\mathbf{M}$ is invertible since

$$\begin{pmatrix} 2 & 2 & 3 \\ 0 & 1 & 3 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 2 \\ 3 & 2 & 2 \end{pmatrix} = \mathbf{I}_3.$$

However, we cannot apply Gaussian elimination since we cannot cancel out the bottom left 1 with a linear combination of 2.

We describe a way to invert over $\mathbb{Z}/m\mathbb{Z}$ with the following proposition.

Proposition 13. *Let* $\mathbf{M} \in (\mathbb{Z}/m\mathbb{Z})^{n\times n}$ *be an invertible matrix, that is,* $\mathbf{M} \in \mathrm{GL}_n(\mathbb{Z}/m\mathbb{Z})$*. Let* $\mathbf{M}'$ *be the inverse of* $\mathbf{M}$ *over the integral domain* $\mathbb{Q}$*. Suppose that*

$$\mathbf{M}' = \begin{pmatrix} \frac{a_{1,1}}{b_{1,1}} & \frac{a_{1,2}}{b_{1,2}} & \cdots & \frac{a_{1,n}}{b_{1,n}} \\ \frac{a_{2,1}}{b_{2,1}} & \frac{a_{2,2}}{b_{2,2}} & \cdots & \frac{a_{2,n}}{b_{2,n}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{n,1}}{b_{n,1}} & \frac{a_{n,2}}{b_{n,2}} & \cdots & \frac{a_{n,n}}{b_{n,n}}, \end{pmatrix}$$

for some non-zero integers $b_{i,j}$ *and integers* $a_{i,j}$*. It follows that*

$$\mathbf{M}^{-1} = \begin{pmatrix} a_{1,1}b_{1,1}^{-1} & a_{1,2}b_{1,2}^{-1} & \cdots & a_{1,n}b_{1,n}^{-1} \\ a_{2,1}b_{2,1}^{-1} & a_{2,2}b_{2,2}^{-1} & \cdots & a_{2,n}b_{2,n}^{-1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1}b_{n,1}^{-1} & a_{n,2}b_{n,2}^{-1} & \cdots & \frac{a_{n,n}}{b_{n,n}}, \end{pmatrix},$$

where the inverses are taken in $\mathbb{Z}/m\mathbb{Z}$*.*

Proof. Let $\mathbf{M} \in GL_n(\mathbb{Z}/m\mathbb{Z})$ and $\tilde{\mathbf{M}}$ the same matrix but considered over the ring $\mathbb{Q}^{n\times n}$. Note that $\mathbf{M}^{-1} \in (\mathbb{Z}/m\mathbb{Z})^{n\times n} \subset \mathbb{Q}^{n\times n}$ and thus $\tilde{\mathbf{M}}$ is also invertible over the ring $\mathbb{Q}$. From Laplace's formula, we deduce that

$$\tilde{\mathbf{M}}^{-1} = \frac{1}{\det \tilde{\mathbf{M}}} \operatorname{adj}(\tilde{\mathbf{M}}), \tag{6}$$

and thus, $\det(\tilde{\mathbf{M}})\tilde{\mathbf{M}}^{-1} = \operatorname{adj}(\tilde{\mathbf{M}})$ has integral coefficients. Therefore, this equality is also true when we take each element modulo m.

That means that $\det(\mathbf{M})\mathbf{M}^{-1} = \operatorname{adj}(\mathbf{M})$, where matrix operations are done over $\mathbb{Z}/m\mathbb{Z}$. Note that $\det(\mathbf{M})$ must be invertible otherwise, it would contradict the fact that $\mathbf{M}$ is invertible. Finally, we can compute

$$\mathbf{M}^{-1} = \det(\mathbf{M})^{-1} \operatorname{adj}(\mathbf{M}),$$

where the inverse is taken over $\mathbb{Z}/m\mathbb{Z}$. Observe that, as desired, the right-hand side of this equation is the same as taking the inverse of $\mathbf{M}$ over $\mathbb{Q}$ and then, transforming the denominator into an inverse modulo m. □

We may ask how to detect if a matrix is invertible over a ring and how many, if there are any, matrices are invertible over a given ring. We give an answer to the first question here and the second one below.

Proposition 14 (Testing Non-Singularity over $\mathbb{Z}/m\mathbb{Z}$**).** *Consider any matrix* $\mathbf{M} \in (\mathbb{Z}/m\mathbb{Z})^{n\times n}$*, we have*

$$\mathbf{M} \in \mathrm{GL}_n(\mathbb{Z}/m\mathbb{Z}) \iff \det \mathbf{M} \in (\mathbb{Z}/m\mathbb{Z})^{\times}.$$

Proof. The ($\Leftarrow$) is a direct result of Equation 6. To prove the other direction ($\Rightarrow$), consider the following equation:

$$\mathbf{M}\mathbf{M}^{-1} = \mathbf{I}_n.$$

By taking the determinant on both sides and using its multiplicative property, we get

$$\det(\mathbf{M}) \det(\mathbf{M}^{-1}) = 1,$$

which by definition, implies that $\det(\mathbf{M})$ is a unit. □

The following result is presented over a prime power residue ring $\mathbb{Z}/p^m\mathbb{Z}$ but it can be extended to more general rings [17].

Proposition 15 (Fraction of Non-Singular Matrices over $(\mathbb{Z}/p^s\mathbb{Z})$). *The ratio of invertible matrices over $(\mathbb{Z}/p^s\mathbb{Z})^{n\times n}$ is*

$$\frac{\prod_{j=1}^{n}(p^n - p^{n-j})}{p^{n^2}}.$$

Observe that this quantity does not depend on s.

Proof. We get from [17, Corollary 2.8] that $\mathrm{GL}_n(\mathbb{Z}/p^s\mathbb{Z}) = p^{(s-1)m^2}\prod_{j=1}^{n}(p^n - p^{n-j})$. If we now divide by the total number of elements in $(\mathbb{Z}/p^s\mathbb{Z})^{n\times n}$, that is $(p^s)^{m^2}$, we get the desired result. □

We argue that most of the matrices over $\mathbb{Z}/p^s\mathbb{Z}$ are invertible. Table 2 displays the ratio for different primes p and values of n. As we can see, they are always a constant fraction of matrices that are invertible and for the most part, it is at least $\frac{1}{2}$.

3.4.2 The Probabilities We first explain intuitively how the weaker structure of the ring makes the work needed to find expressions for the probabilities more challenging. Consider any field $\mathbb{Z}/p\mathbb{Z}$ and any non-zero integer in $x \in \mathbb{Z}/p\mathbb{Z}$. It follows that the number of pairs of non-zero elements in $\mathbb{Z}/p\mathbb{Z}$ such that their product is x does not depend on x. This fact can be used when we want to count how many pairs of vectors with non-zeroes entries have a non-zero inner product. In fact, this is used for the expressions describing the probabilities given in Proposition 11.

On the other hand, this is not true in general over integer residue rings as depicted by the next example.

Example 4. Consider the ring $\mathbb{Z}/6\mathbb{Z}$, which is not a field. We display below its multiplication tables and color the non-zero entries such that the same values have the same colors.

s \ p	2	3	5	7	11	13	17	19	23	29
1	0.500	0.667	0.800	0.857	0.909	0.923	0.941	0.947	0.957	0.966
2	0.375	0.593	0.768	0.840	0.902	0.918	0.938	0.945	0.955	0.964
3	0.328	0.571	0.762	0.837	0.901	0.917	0.938	0.945	0.955	0.964
4	0.308	0.564	0.761	0.837	0.901	0.917	0.938	0.945	0.955	0.964
5	0.298	0.561	0.760	0.837	0.901	0.917	0.938	0.945	0.955	0.964
6	0.293	0.561	0.760	0.837	0.901	0.917	0.938	0.945	0.955	0.964
7	0.291	0.560	0.760	0.837	0.901	0.917	0.938	0.945	0.955	0.964
8	0.290	0.560	0.760	0.837	0.901	0.917	0.938	0.945	0.955	0.964
9	0.289	0.560	0.760	0.837	0.901	0.917	0.938	0.945	0.955	0.964
10	0.289	0.560	0.760	0.837	0.901	0.917	0.938	0.945	0.955	0.964
11	0.289	0.560	0.760	0.837	0.901	0.917	0.938	0.945	0.955	0.964

Table 2: Fraction of invertible matrices over $(\mathbb{Z}/p^s\mathbb{Z})^{n\times n}$. Note that the value does not depend on s.

$\times$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	1	0	5

Looking at the multiplication table we see that the number of pairs that multiply to 1 (colored in red) is two, whereas there are five pairs that multiply to 2.

This motivates the expression below. We use the following recursive expression to compute the number of pairs of vectors of length j with non-zero entries that yield an inner product different from x.

Definition 21. *Let m and j be strictly positive integers and $x \in \mathbb{Z}/m\mathbb{Z}$. Then we let*

$$\mathfrak{I}(m,x,j) = |\{(\mathbf{u},\mathbf{v}) \in (((\mathbb{Z}/m\mathbb{Z}) \setminus \{0\})^n)^2 \mid \mathbf{u}\mathbf{v}^\top \neq x \pmod m\}|.$$

The notation $\mathfrak{I}$ stands for the "i" in inner product. We can also define $\mathfrak{I}$ recursively as

$$\mathfrak{I}(m,x,j) = \begin{cases} \mathbb{1}_{x\not\equiv 0 \pmod m} & \text{if } j = 0, \\ \sum_{w_h=1}^{m-1}\sum_{w_e=1}^{m-1} \mathfrak{I}(m, x - w_h w_e \pmod m, j-1) & \text{otherwise.} \end{cases}$$

We can think of the recursive calls as follows: We test all combinations of the pair (w_h, w_e) that correspond to the j index and recursively calls $\mathfrak{I}$ by changing the non-desired value x to $x + w_e w_h \pmod m$.

We use $\mathfrak{I}$ in order to introduce another useful combinatorics notation. Given two strictly positive integers t and w, it counts the number of pairs of vectors of respective weight w and t such that their inner product is not x. As we will see later, it will be very helpful when constructing an expression for the probabilities.

Definition 22. *Let m, x, n, t and w be integers. We define the following quantity:*

$$\begin{aligned}&\tilde{\mathfrak{I}}(m, x, n, w, t)\\ &= |\{(\mathbf{u}, \mathbf{v}) \in S^n_{m,H}(\mathbf{0}, t) \times S^n_{m,H}(\mathbf{0}, w) \mid \mathbf{u}\mathbf{v}^\top \not\equiv x \pmod m\}|,\end{aligned}$$

where spheres are considered over $\mathbb{Z}/m\mathbb{Z}$.

Using Equation 21 and binomial coefficients we can easily find an expression for $\tilde{\mathfrak{I}}$.

Proposition 16. *Let m, x, n, t and w be integers such that $t \leq n$, $w \leq n$ and $x \in \mathbb{Z}/m\mathbb{Z}$, then*

$$\begin{aligned}&\tilde{\mathfrak{I}}(m, x, n, w, t)\\ &\quad= \sum_{j=0}^{\nu} \mathfrak{I}(m, x, j) \binom{n}{j} \binom{n-j}{\mu-j} \binom{n-\mu}{\nu-j} (m-1)^{\mu+\nu-2j},\end{aligned} \tag{7}$$

where $\nu \triangleq \min(t, w)$ and $\mu \triangleq \max(t, w)$.

Proof. This problem reduces nicely to the one solved by the function $\mathfrak{I}(m, x, j)$. We let ν and μ be as in the proposition. We partition the set

$$S = \{(\mathbf{u}, \mathbf{v}) \in ((\mathbb{Z}/m\mathbb{Z})^n)^2 \mid \mathrm{wt}_H(\mathbf{u}) = t \wedge \mathrm{wt}_H(\mathbf{v}) = w \wedge \mathbf{u}\mathbf{v}^\top \not\equiv x \pmod m\}$$

into $\nu+1$ subsets $(S_j)_{j\in\{0,\dots,\nu\}}$ such that $S_j = \{(\mathbf{u}, \mathbf{v}) \mid |\mathrm{Supp}(\mathbf{x}) \cap \mathrm{Supp}(\mathbf{y})| = j\}$.

We can decompose $|S_j|$ in three parts.

1. The number of ways we can arrange these j non-zeroes common entries within the two vectors times;
2. The number of ways we can dispose the others;
3. How many combinations of two vectors of size j with non-zeroes entries have an inner product different from x.

The first and last values are respectively $\binom{n}{j}$ and $\mathfrak{I}(m, x, j-1)$. The second value factor is $\binom{n-j}{\mu-j}\binom{n-\mu}{\nu-j}(m-1)^{\mu+\nu-2j}$. The right part reflects the fact that if the $\mathbf{u}_i = 0$ then $\mathbf{v}_i$ will not impact the inner product and the left part counts how we can separate the remaining non-zeroes entries so they do not coincide. □

Now we can translate the equations for the probabilities $\tilde{p}^{++}_{w,t,q}$ and $\tilde{q}^{++}_{w,t,q}$ into the more general ring $\mathbb{Z}/m\mathbb{Z}$ case. We let $\mathcal{E}_t$ be the set of vectors in $(\mathbb{Z}/m\mathbb{Z})^n$ of Hamming weight t, that is, the set of possible error vectors.

Definition 23. *Let m be an integer, we define the two following probabilities:*

$$\tilde{p}^{++}_{w,t,m} = \Pr_{\substack{\mathbf{h}\in\mathcal{C}^{\perp}_{w}\\ \mathbf{e}\in\mathcal{E}_t}}[\mathbf{h}_i \neq 0 \mid \mathbf{e}_i \neq 0 \wedge \mathbf{h}\mathbf{y}^{\top} \neq 0],$$

$$\tilde{q}^{++}_{w,t,m} = \Pr_{\substack{\mathbf{h}\in\mathcal{C}^{\perp}_{w}\\ \mathbf{e}\in\mathcal{E}_t}}[\mathbf{h}_i \neq 0 \mid \mathbf{e}_i = 0 \wedge \mathbf{h}\mathbf{y}^{\top} \neq 0].$$

Observe that we have to consider the error itself since we are working over a ring. This is why the probabilities are also taken with respect to $\mathbf{e}$ in $\mathcal{E}_t$. Using the function $\tilde{\mathfrak{J}}$ introduced in Definition 22, we can write an expression of these probabilities.

Proposition 17. *Let w and t be integers such that $0 \leq w, t \leq n$, then*

$$\tilde{p}^{++}_{w,t,m} = \frac{\sum_{w_h=1}^{n}\sum_{w_e=1}^{n}\tilde{\mathfrak{J}}(m, w_h w_e, n', w', t')}{(m-1)\tilde{\mathfrak{J}}(m,0,n',w,t') + \sum_{w_h=1}^{n}\sum_{w_e=1}^{n}\tilde{\mathfrak{J}}(m, w_h w_e, n', w', t')},$$

where $n' \triangleq n-1$, $w' \triangleq w-1$ and $t' \triangleq t-1$. Also, for any integer $t < n$, we have

$$\tilde{q}^{++}_{w,t,m} = \frac{\sum_{w_h=1}^{n}\tilde{\mathfrak{J}}(m,0,n',w',t)}{\tilde{\mathfrak{J}}(m,0,n',w,t) + \sum_{w_h=1}^{n}\tilde{\mathfrak{J}}(m,0,n',w',t)}.$$

In any other case, these probabilities are zeroes.

Proof. We prove the statement for $\tilde{p}^{++}_{w,t,m}$, the other can be proved similarly. Without loss of generality, we will assume that $i = 1$. The numerator counts the number of pairs of vector $(\mathbf{e}, \mathbf{h})$, such that $\mathbf{e}_1 \neq 0$, $\mathbf{h}_1 \neq 0$ and $\mathbf{h}\mathbf{y}^{\top} = \mathbf{h}\mathbf{e}^{\top} \neq 0$. This is done through the helper function $\tilde{\mathfrak{J}}(m, x, n', w', t')$. We let $n' = n-1$, $w' = w-1$, and $t' = t-1$ to account for the fact that we have fixed $\mathbf{h}_1$ and $\mathbf{e}_1$ to be non-zero and $x = w_h w_e$ to set the current value of the inner product.

Likewise for the denominator, we count the number of pairs of vectors such that $\mathbf{e}_1 \neq 0$ and $\mathbf{h}\mathbf{y}^{\top} = \mathbf{h}\mathbf{e}^{\top} \neq 0$. The numerator is a special denominator case, so it appears on the right. The remaining possibilities are the ones with $\mathbf{h}_1 = 0$. In this case, the $\mathbf{e}_1$ value does not impact the inner product; thus, there are $m-1$ identic possibilities. The parameters of N are set so we decrease the Hamming weight of $\mathbf{e}$ and keep the weight for $\mathbf{h}$. □

Finally, we can apply the analysis from [34] to calculate the approximate size of $\mathcal{H}_w$ to achieve 95% of successful decoding. Namely, we have the following proposition:

Proposition 18 (Cardinality of $\mathcal{H}_w$ over $\mathbb{Z}/m\mathbb{Z}$). *Over $\mathbb{Z}/m\mathbb{Z}$, in order for statistical decoding to achieve 95% probability of success we need $\mathcal{H}_w$ to be of the following size:*

$$|\mathcal{H}_w| \approx 2.72\frac{m}{m-1}\tilde{p}^{++}_{w}\frac{1-\tilde{p}^{++}_{w}}{\left(\tilde{p}^{++}_{w} - \tilde{q}^{++}_{w}\right)^2}. \tag{8}$$

Note that this expression is close to the one presented in Proposition 12, the only difference being in the probabilities. The next section is devoted to comparing the cardinality of $\mathcal{H}_w$ for different rings and fields.

3.4.3 Comparison with Fields We first perform an empirical comparison and then verify our observations by analyzing the theoretical values. For the following comparisons, we will focus on $\mathbb{Z}/m\mathbb{Z}$ for $m \in \{13, 15, 17\}$. Note that 13 and 17 are both prime and therefore the case of the ring $\mathbb{Z}/15\mathbb{Z}$ is sandwiched between the two fields $\mathbb{Z}/13\mathbb{Z}$ and $\mathbb{Z}/17\mathbb{Z}$.

For each experiment, we pick a random generator 4×20 matrix of a code with distance $d = 9$ and set the Hamming weight of the error, t, to be 4 and $w = 9$. We use different set $\mathcal{H}_w$ of repsective cardinality 5, $5 \cdot 10$, $5 \cdot 10^2$ and $5 \cdot 10^3$. We repeat each experiment 1000 times. The results are depicted in the left part of Table 3. We have colored in red the frequencies that achieved the 95% threshold.

$\lvert\mathcal{H}_w\rvert$	metric	m	Success Frequency	$\lvert\mathcal{H}_w\rvert$	metric	m	Success Frequency
5	Hamming	13	0.431	5	Lee	13	0.488
ß 5	Hamming	15	0.464	5	Lee	15	0.499
5	Hamming	17	0.409	5	Lee	17	0.502
$5 \cdot 10$	Hamming	13	0.589	$5 \cdot 10$	Lee	13	0.672
$5 \cdot 10$	Hamming	15	0.623	$5 \cdot 10$	Lee	15	0.672
$5 \cdot 10$	Hamming	17	0.608	$5 \cdot 10$	Lee	17	0.651
$5 \cdot 10^2$	Hamming	13	0.925	$5 \cdot 10^2$	Lee	13	0.956
$5 \cdot 10^2$	Hamming	15	0.933	$5 \cdot 10^2$	Lee	15	0.958
$5 \cdot 10^2$	Hamming	17	0.931	$5 \cdot 10^2$	Lee	17	0.952
$5 \cdot 10^3$	Hamming	13	1.000	$5 \cdot 10^3$	Lee	13	1.000
$5 \cdot 10^3$	Hamming	15	1.000	$5 \cdot 10^3$	Lee	15	1.000
$5 \cdot 10^3$	Hamming	17	1.000	$5 \cdot 10^3$	Lee	17	1.000

Table 3: Success frequency of statistical decoding computed over 1000 samples for different integer residue rings over the Hamming and Lee metrics and for sets $\mathcal{H}_w$ of different cardinalities. The frequencies that achieved the desired 95% probability of success are colored in red.

As expected, the success probability increase with the size of $\mathcal{H}_w$. In the field case, it appears that the success probability has a low dependence on the size of the space. Note that in the ring case, the success probability tends to be better. We now confirm these observations by presenting the graph of the theoretical values of $|\mathcal{H}_w|$ to obtain the desired success probability.

Figure 5 shows the different cardinalities of $\mathcal{H}_w$ for different integer residue rings $\mathbb{Z}/m\mathbb{Z}$, where m is represented in the abscissas. We have colored in red the points that correspond to a field (when m is a prime number). As one could expect from the example, the cardinality required in the field case is much

higher than in the ring case. In other words, by moving from a field to a ring of comparable size, it appears that the statistical decoding performs better. Observe also that as m increases, the required size seems to decrease.

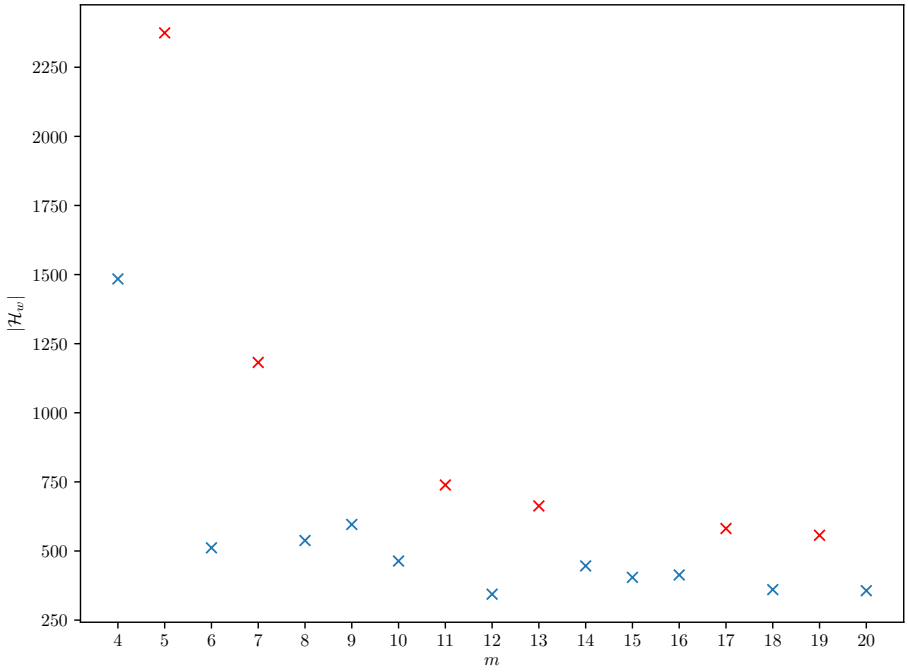


Fig. 5: Cardinality of $\mathcal{H}_w$ for different integer residue rings over the Hamming metric. When m is prime the points are colored in red whereas if m is composite, there are colored in blue.

We now compare rings and fields of the same size. For the following experiment, we let $d = 7$, $t = 3$, and $w = 9$. We consider the rings $\mathbb{Z}/4\mathbb{Z}$, $\mathbb{Z}/8\mathbb{Z}$, $\mathbb{Z}/16\mathbb{Z}$ and the fields $\mathbb{F}_4$, $\mathbb{F}_8$, $\mathbb{F}_{16}$. Observe that these structures are nonisomorphic and that we can arrange them into pairs of the same cardinality. Table 4 presents the different empirical success probabilities over the Hamming metric computed with 1000 iterations. As in Table 3, We have colored in red the probability of success that has reached 95%. Looking at the results, it seems that statistical decoding is more efficient over the integer residue ring than over a field of identical size. We also see that increasing the structure size improves the success probability.

3.4.4 Simplier expression for $\mathfrak{I}$ over $\mathbb{Z}/2^s\mathbb{Z}$ Although the computation of running time is not necessary when decrypting a random code, it is crucial

$\lvert\mathcal{H}_w\rvert$	Ambient Space	Success Frequency	$\lvert\mathcal{H}_w\rvert$	Ambient Space	Success Frequency
5	$\mathbb{Z}/4\mathbb{Z}$	0.584	5	$\mathbb{F}_4$	0.530
5	$\mathbb{Z}/8\mathbb{Z}$	0.627	5	$\mathbb{F}_8$	0.598
5	$\mathbb{Z}/16\mathbb{Z}$	0.622	5	$\mathbb{F}_{16}$	0.618
$5 \cdot 10$	$\mathbb{Z}/4\mathbb{Z}$	0.785	$5 \cdot 10$	$\mathbb{F}_4$	0.744
$5 \cdot 10$	$\mathbb{Z}/8\mathbb{Z}$	0.828	$5 \cdot 10$	$\mathbb{F}_8$	0.873
$5 \cdot 10$	$\mathbb{Z}/16\mathbb{Z}$	0.863	$5 \cdot 10$	$\mathbb{F}_{16}$	0.896
$5 \cdot 10^2$	$\mathbb{Z}/4\mathbb{Z}$	0.978	$5 \cdot 10^2$	$\mathbb{F}_4$	0.873
$5 \cdot 10^2$	$\mathbb{Z}/8\mathbb{Z}$	0.996	$5 \cdot 10^2$	$\mathbb{F}_8$	0.999
$5 \cdot 10^2$	$\mathbb{Z}/16\mathbb{Z}$	0.997	$5 \cdot 10^2$	$\mathbb{F}_{16}$	1.000
$5 \cdot 10^3$	$\mathbb{Z}/4\mathbb{Z}$	1.000	$5 \cdot 10^3$	$\mathbb{F}_4$	1.000
$5 \cdot 10^3$	$\mathbb{Z}/8\mathbb{Z}$	1.000	$5 \cdot 10^3$	$\mathbb{F}_8$	1.000
$5 \cdot 10^3$	$\mathbb{Z}/16\mathbb{Z}$	1.000	$5 \cdot 10^3$	$\mathbb{F}_{16}$	1.000

Table 4: Success frequency of statistical decoding computed over 1000 samples for different integer residue rings and finite fields over the Hamming for sets $\mathcal{H}_w$ of different cardinalities. The frequencies that achieved the desired 95% probability of success are colored in red.

for determining the size of the public key. The most natural way to do this is to iterate over larger and larger parameters, stopping when the complexity reaches the given security threshold. Furthermore, numerical results about the complexity can help to analyze and compare statistical decryption. Unfortunately, the current expression for $\mathfrak{I}$ has a quadratic number of sub-calls. If we had to implement this function, we would end up with a lot of overhead due to the large number of sub-calls. In this section, we derive a simpler formula that has only a log-linear number of sub-calls in the special case of the ring $\mathbb{Z}/2^s\mathbb{Z}$. We let $q \triangleq 2^s$ and recall the recursive expression for $\mathfrak{I}(q, x, j)$.

$$\mathfrak{I}(q,x,j) = \begin{cases} \mathbb{1}_{x \not\equiv 0 \pmod q} & \text{if } j = 0, \\ \sum_{w_h=1}^{q-1} \sum_{w_e=1}^{q-1} \mathfrak{I}(q, x + w_h w_e, j-1) & \text{otherwise.} \end{cases}$$

Observe that the only part that changes within the sub-calls is the second parameter of $\mathfrak{I}$, namely $x + w_h w_e$. This parameter can only take values within $\mathbb{Z}/q\mathbb{Z}$ while we do q^2 sub-calls. By the pigeonhole principle, some calls must be redundant. By counting how many calls with the same parameter $x + w_h w_e$ appear we can reduce the number of sums from two to one. Note that we have to take special care of the case $x + w_h w_e = x$ because w_h and w_e cannot be zero. We prove the following proposition about the distribution of the product of two elements of $\mathbb{Z}/q\mathbb{Z}$.

Proposition 19. *Let X and Y be independent and uniform random variables over $\mathbb{Z}/q\mathbb{Z}$, $Z = XY$ their product over $\mathbb{Z}/q\mathbb{Z}$, $k \in \mathbb{Z}/2^s\mathbb{Z}$ and $\delta_{s-1}, \ldots, \delta_1, \delta_0$ and $\delta'_{s-1}, \ldots, \delta'_1, \delta'_0$ the respective binary representations of k and $k-1$[1], then*

$$\Pr[Z = k] = q^{-1}\left(1 + 2\sum_{i=0}^{m-1}(\delta'_i - \delta_i)\right).$$

Proof. The proof is a direct result of [42, Equation 1.2]. We state the result here for completeness:

$$\Pr[Z \leq k] = (k+1)q^{-1} + 2q^{-1}\sum_{i=0}^{m-1}(1 - \delta_i).$$

From this point, we can mechanically compute the desired result, namely

$$\begin{aligned}\Pr[Z = k] &= \Pr[Z \leq k] - \Pr[Z \leq k-1] \\ &= q^{-1}\left(1 + 2\sum_{i=0}^{m-1}((1-\delta_i) - (1-\delta'_i))\right) \\ &= q^{-1}\left(1 + 2\sum_{i=0}^{m-1}(\delta'_i - \delta_i)\right).\end{aligned}$$

□

Notice that for any non-zero $k \in \mathbb{Z}/q\mathbb{Z}$, we have

$$|\{(x,y) \in (\mathbb{Z}/q\mathbb{Z} \setminus \{0\})^2 \mid xy = k\}| = \Pr[Z = k]q^2. \tag{9}$$

and for zero,

$$\begin{aligned}|\{(x,y) \in (\mathbb{Z}/q\mathbb{Z} \setminus \{0\})^2 \mid xy = 0\}| &= \Pr[Z = 0]q^2 - (2q-1) \\ &= \Pr[Z = 0]q^2 - 2q + 1. \end{aligned} \tag{10}$$

Finally, we get our more efficiently computable formula for $\mathfrak{I}$.

Theorem 11. *Over $\mathbb{Z}/q\mathbb{Z}$, for any integer positive j and $x \in \mathbb{Z}/q\mathbb{Z}$, we have*

$$\mathfrak{I}(q,x,j) = \begin{cases} \mathbb{1}_{x \not\equiv 0 \pmod q} & \text{if } j = 0, \\ \left(\Pr[Z=0]q^2 - 2q + 1\right)\mathfrak{I}(2^s, x, j-1) & \\ + \sum_{k=1}^{q-1}\Pr[Z=k]q^2\mathfrak{I}(q, x-k \pmod q, j-1) & \text{otherwise.} \end{cases}$$

Proof. This expression is different from the one presented in Definition 21 only if $j > 0$ and thus we will consider only the second case. For the following equations,

[1] With the least significant bit being the right most bit.

let $\bar{\mathbf{u}} \triangleq \mathbf{u}_{[\![n]\!]\setminus\{i\}}$ and $\bar{\mathbf{v}} \triangleq \mathbf{v}_{[\![n]\!]\setminus\{i\}}$ and assume that all computation done inside the set bracket notation are over $\mathbb{Z}/q\mathbb{Z}$. We have by definition that

$$
\begin{aligned}
\mathfrak{I}(q,x,j) &= |\{(\mathbf{u},\mathbf{v}) \in (((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^n)^2 \mid \mathbf{u}\mathbf{v}^\top \neq x\}| \\
&= |\bigcup_{k=0}^{q-1} \{(\mathbf{u},\mathbf{v}) \in (((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^n)^2 \mid \mathbf{u}_i\mathbf{v}_i = k \wedge \bar{\mathbf{u}}\bar{\mathbf{v}}^\top \neq x-k\}| \\
&= \sum_{k=0}^{q-1} |\{(\mathbf{u},\mathbf{v}) \in (((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^n)^2 \mid \mathbf{u}_i\mathbf{v}_i = k \wedge \bar{\mathbf{u}}\bar{\mathbf{v}}^\top \neq x-k\}| \\
&= |\{(\mathbf{u},\mathbf{v}) \in (((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^n)^2 \mid \mathbf{u}_i\mathbf{v}_i = 0 \wedge \bar{\mathbf{u}}\bar{\mathbf{v}}^\top \neq x\}| \\
&+ \sum_{k=1}^{q-1} |\{(\mathbf{u},\mathbf{v}) \in (((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^n)^2 \mid \mathbf{u}_i\mathbf{v}_i = k \wedge \bar{\mathbf{u}}\bar{\mathbf{v}}^\top \neq x-k\}| \\
&= |\{(\mathbf{u}_i,\mathbf{v}_i) \in ((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^2 \mid \mathbf{u}_i\mathbf{v}_i = 0\}| \\
&\cdot |\{(\mathbf{u},\mathbf{v}) \in \left(((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^{n-1}\right)^2 \mid \bar{\mathbf{u}}\bar{\mathbf{v}}^\top \neq x\}| \\
&+ \sum_{k=1}^{q-1} \Bigg(|\{(\mathbf{u}_i,\mathbf{v}_i) \in ((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^2 \mid \mathbf{u}_i\mathbf{v}_i = k\}| \\
&\cdot |\{(\mathbf{u},\mathbf{v}) \in \left(((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^{n-1}\right)^2 \mid \bar{\mathbf{u}}\bar{\mathbf{v}}^\top \neq x-k\}| \Bigg) \\
&= |\{(\mathbf{u}_i,\mathbf{v}_i) \in ((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^2 \mid \mathbf{u}_i\mathbf{v}_i = 0\}| \mathfrak{I}(q,x,j-1) \\
&+ \sum_{k=1}^{q-1} \Bigg(|\{(\mathbf{u}_i,\mathbf{v}_i) \in ((\mathbb{Z}/q\mathbb{Z})\setminus\{0\})^2 \mid \mathbf{u}_i\mathbf{v}_i = k\}| \\
&\cdot \mathfrak{I}(q, x-k \pmod q, j-1) \Bigg).
\end{aligned}
$$

By substituting Equations 10 and 9, we get the desired result. □

Another improvement would be to use memoization or a bottom-up approach in order to avoid recomputing the same value twice. We could argue that there still has two imbricated loops but in this new expression, the inner loop is exponentially smaller than in the previous expression. Moreover, with $\mathcal{O}(q) = \mathcal{O}(2^s)$ memory we could precompute all values of $\Pr[Z = k]$ and store them in a look-up table as a way of removing any need for an inner sum and thus, achieving linear complexity.

3.5 Statistical Decoding in the Lee metric

In this section, we look at how changing the metric while staying in a field affects statistical decoding. The algorithm over the Hamming metric naturally extends over the Lee metric. In fact, the statistical bias depends on the Hamming weight

of the error vector and not on the values of its non-zero entries. Note that for a given Lee weight, many vectors of different Hamming weights can achieve this Lee weight. This leads us to find an expression for the distribution of Hamming weight of a uniformly distributed error vector of a given Lee weight. We provide a non-closed expression for this distribution and its expectation. Finally, we empirically compare the success probability of codes over $\mathbb{Z}/m\mathbb{Z}$ in the Hamming and Lee metric.

3.5.1 Hamming Weight's Distribution of the Error For clarity, we will assume in the following results that m is odd. Similar results can be derived in the even case. The Hamming weight of a vector sampled uniformly in all vectors of a given Lee weight is a random variable. We are interested in finding its probability mass function. When the given Lee weight is t, the length of the vector is n and the ring is $\mathbb{Z}/m\mathbb{Z}$, we use the notation $p_{t,n,m}(k)$ for the probability that the Hamming weight is k. We have the following result:

Proposition 20 (Error's Hamming Weight Distribution). *Supposing that* $\mathbf{e}$ *is uniformly distributed within the set of vectors of Lee weight* t*, then*

$$p_{t,n,m}(k) \triangleq \Pr[wt_H(\mathbf{e}) = k] = \frac{2^k \binom{n}{k} C(t,k,\lfloor m/2 \rfloor)}{\sum_{\ell=0}^{n} 2^\ell \binom{n}{\ell} C(t,\ell,\lfloor m/2 \rfloor)}.$$

Proof. The number of ways we can have a vector of Lee weight t and Hamming weight ℓ is $2^\ell \binom{n}{\ell} C(t,\ell,\lfloor m/2 \rfloor)$. The factor $\binom{n}{\ell}$ counts the number of ways we can distribute the ℓ non-zero entries in the vector, the factor $C(t,\ell,\lfloor m/2 \rfloor)$ counts the number of ways we can add up to the correct Lee weight t using only elements between 1 and $\lfloor \frac{m}{2} \rfloor$. Finally, the term 2^ℓ accounts for that for each non-zero element $\mathbf{e}_i \leq \lfloor \frac{m}{2} \rfloor$, we can choose its opposite element $m - \mathbf{e}_i$, of identic Lee weight. The opposite element is distinct since we assumed that m is odd. To conclude, note that the numerator of the probability counts the number of vectors having Lee weight t and Hamming weight k and that the denominator is the size of all possible vectors of Lee weight t. □

By definition, the expected value of $\mathrm{wt}_H(\mathbf{e})$ is

$$\begin{aligned}\mathbb{E}[\mathrm{wt}_H(\mathbf{e})] &\triangleq \sum_{k=1}^{n} k \Pr[\mathrm{wt}_H(\mathbf{e}) = k] \\ &= \sum_{k=1}^{n} \frac{k 2^k \binom{n}{k} C(t,k,\lfloor m/2 \rfloor)}{\sum_{\ell=0}^{n} 2^\ell \binom{n}{\ell} C(t,\ell,\lfloor m/2 \rfloor)}, \end{aligned} \tag{11}$$

and its variance is

$$\begin{aligned}\mathbb{V}[\mathrm{wt}_H(\mathbf{e})] &\triangleq \mathbb{E}[\mathrm{wt}_H(\mathbf{e})^2] - \mathbb{E}[\mathrm{wt}_H(\mathbf{e})]^2 \\ &= \sum_{k=1}^{n} \frac{k^2 2^k \binom{n}{k} C(t,k,\lfloor m/2 \rfloor)}{\sum_{\ell=0}^{n} 2^\ell \binom{n}{l} C(t,\ell,\lfloor m/2 \rfloor)} - \\ &\left[\sum_{k=1}^{n} \frac{k^2 2^k \binom{n}{k} C(t,k,\lfloor m/2 \rfloor)}{\sum_{\ell=0}^{n} 2^\ell \binom{n}{\ell} C(t,\ell,\lfloor m/2 \rfloor)}\right]^2. \end{aligned} \tag{12}$$

Consider the exact SDP over the Lee metric and error vectors weight t. We show by an example that with a very small probability, solving this problem with statistical decoding reduces to solving the same problem over the Hamming metric with an error of Hamming weight t. In other words, using statistical decoding, the exact Lee-SDP reduces, with some probability, to a simpler instance (in terms of the weight of the error) of the exact Hamming-SDP.

Example 5 (Hamming weight's distribution). Consider the Hamming weight's distribution of uniformly distributed vectors of Lee weight equals to 20. Its probability mass function, computed with Proposition 20, is shown in Figure 6. In this example, the base ring is $\mathbb{Z}/21\mathbb{Z}$ and $n = 25$. From the graph, it appears that there is no vector of Lee weight equal to 20, which is wrong. The probability of getting such a vector is really small ($\approx 6.25 \cdot 10^{-6}$) and thus is not visible in the graph. Therefore, with very high probability, the statistical decoding problem over the Lee metric reduces another statistical decoding instance over the Hamming metric with a smaller t.

The next result state that indeed, the distribution is mostly concentrated on its expectation. Show that numerically using Chebyshev's inequality. We recall this inequality in the following theorem:

Theorem 12 (Chebyshev's Inequality [30]). *Let X be any random variable with finite variance σ^2 and finite expectation μ Then for any $\lambda > 0$, we have*

$$\Pr[|X - \mu| \geq \lambda] \leq \frac{\sigma^2}{\lambda^2}.$$

As required by this inequality, Equations 11 and 12 are finite. By setting $X = \mathrm{wt}_H(\mathbf{e})$ and $\lambda = \frac{1}{3}\mathbb{E}[\mathrm{wt}_H(\mathbf{e})]$ we can upper bound the probabilty that $\mathrm{wt}_H(\mathbf{e})$ is not within the interval $[\mu - \frac{1}{3}\mu; \mu + \frac{1}{3}\mu]$, where $\mu = \mathbb{E}[\mathrm{wt}_H(\mathbf{e})]$. We compute numerically this upper bounds for $n \in \{50, 55, \ldots, 60\}$ and $t \in \{10, 15, \ldots, 40\}$ over $\mathbb{Z}/21\mathbb{Z}$ in Table 5. As we would expect, the probability is low and thus, the distribution of the Hamming weight is mostly concentrated on its expectation.

Figure 7 shows the expected Hamming weight and its standard deviation. This calculation is performed for different values of n. The Lee weight is displayed in red as a reference and the computation is done over the ring $\mathbb{Z}/21\mathbb{Z}$, as in the example above. Note that the Hamming weight does not go beyond n since it makes no sense to have a vector of length n having Hamming weight strictly

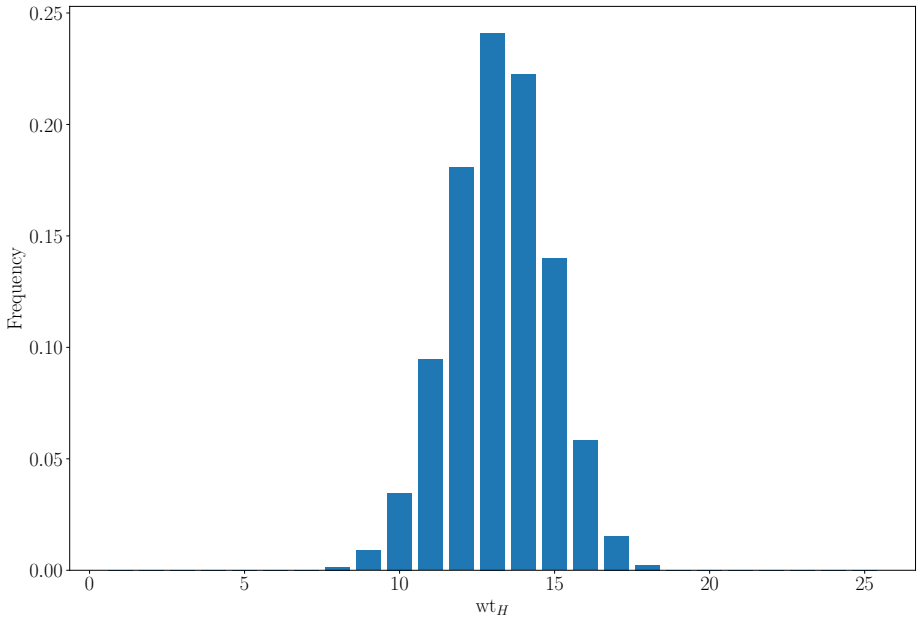


Fig. 6: Hamming weight's distribution of a vector of Lee weight 20 and length 25 over $\mathbb{Z}/21\mathbb{Z}$. The values are computed using Proposition 20.

larger than n. This figure illustrates two phenomena concerning the discrepancy between the number of erroneous positions, also known as the Hamming weight and the Lee weight:

1. The gap grows as t grows;
2. The gap shrinks as n grows.

In other words, it seems that as t increases, the statistical decoding performs better on the Lee metric than on its Hamming counterpart.

On the other hand, we could ask ourselves what impact has m on the distribution. Figure 8 depicts the expected Hamming weight under different values of m and $n = 50$. As explained in Remark 2, the case $m = 3$ is equivalent to the Hamming metric, and the blue plot is the identity function. By increasing m, it seems that we converge to something that is close to the green plot. Especially, the size of the base field has little impact on the expected Hamming weight.

3.6 Statistical Decoding over $\mathbb{Z}/m\mathbb{Z}$ in the Lee metric

It seems that replacing the field with a ring and changing to the Lee metric improves the success probability of the algorithm. Putting the pieces together, we can expect that statistical decoding over the ring $\mathbb{Z}/m\mathbb{Z}$ in the Lee metric has both advantages. In this section, we show empirically that our hypothesis is true.

t \ n	50	55	60	65	70	75	80	85	90	95	100
10	0.080	0.073	0.067	0.062	0.057	0.054	0.050	0.047	0.045	0.042	0.040
15	0.081	0.074	0.068	0.063	0.059	0.055	0.052	0.049	0.046	0.044	0.042
20	0.080	0.073	0.068	0.063	0.059	0.055	0.052	0.049	0.046	0.044	0.042
25	0.078	0.072	0.067	0.062	0.058	0.055	0.052	0.049	0.046	0.044	0.042
30	0.075	0.070	0.065	0.061	0.057	0.054	0.051	0.048	0.046	0.044	0.042
35	0.072	0.067	0.063	0.060	0.056	0.053	0.050	0.048	0.045	0.043	0.041
40	0.069	0.065	0.061	0.058	0.055	0.052	0.049	0.047	0.045	0.043	0.041

Table 5: Probability that an error vector $\mathbf{e}$ of Lee weight t and length n over $\mathbb{Z}/21\mathbb{Z}$ has not an Hamming weight in the interval $[\mu - \frac{1}{3}\mu; \mu + \frac{1}{3}\mu]$, where $\mu = \mathbb{E}[\mathrm{wt}_H(\mathbf{e})]$.

The right part of Table 3 presents the success probability over 1000 experiments under different rings in the Lee metric. The parameters are the same as in the left part of the table so we can compare how changing the metric affects the empirical success probability. As expected, there is a general improvement in the probability in the Lee metric compared to the Hamming metric. It is worth noticing that when m is a prime *i.e.* when the structure is a field, the improvement is more significant.

We conclude this section by giving an explicit expression for the size of $|\mathcal{H}_w|$ to achieve the 95% success probability in the ring case embedded with the Lee metric. First, we translate the probabilities $\tilde{p}^{++}_{w,t,m}$ and $\tilde{p}^{++}_{w,t,m}$ over the ring $\mathbb{Z}/m\mathbb{Z}$.

Definition 24. *Let $\mathcal{E}_t \subseteq (\mathbb{Z}/m\mathbb{Z})^n$ be the set of vectors of Hamming weight t.*

$$\bar{p}^{++}_{w,t,m} = \Pr_{\substack{\mathbf{h}\in\mathcal{C}^{\perp}_w \\ \mathbf{e}\in\mathcal{E}_t}}[\mathbf{h}_i \neq 0 \mid \mathbf{e}_i \neq 0 \wedge \mathbf{h}\mathbf{y}^{\top} \neq 0],$$

$$\bar{q}^{++}_{w,t,m} \triangleq \Pr_{\substack{\mathbf{h}\in\mathcal{C}^{\perp}_w \\ \mathbf{e}\in\mathcal{E}_t}}[\mathbf{h}_i \neq 0 \mid \mathbf{e}_i = 0 \wedge \mathbf{h}\mathbf{y}^{\top} \neq 0].$$

We get the following expression using the law of total probability and Proposition 20.

Proposition 21. *We can describe the probabilities $\bar{p}^{++}_{w,t,m}$ and $\bar{q}^{++}_{w,t,m}$ as follows:*

$$\bar{p}^{++}_{w,t,m} = \sum_{k=1}^{\min\{n,t\}} p_{t,n,m}(k) \cdot \tilde{p}^{++}_{w,k,m},$$

$$\bar{q}^{++}_{w,t,m} = \sum_{k=1}^{\min\{n,t\}} p_{t,n,m}(k) \cdot \tilde{q}^{++}_{w,k,m}.$$

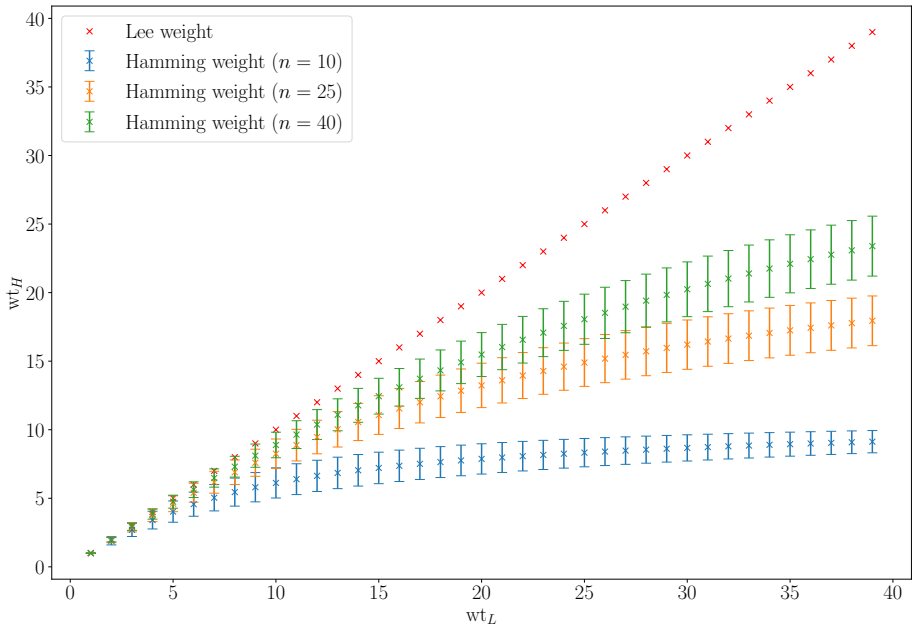


Fig. 7: Expected Hamming weight together with its standard deviation of a random vector of a given Lee weight (represented in the abscissa) over $\mathbb{Z}/21\mathbb{Z}$. There are three plots for three different n.

Figure 9 displays the theoretical values of the cardinality of $\mathcal{H}_w$ in order to achieve the desired success probability over the ring $(\mathbb{Z}/m\mathbb{Z})^{30}$ for the Hamming metric (blue) and the Lee metric (orange). The parameters are set as follows: $t = 7$ and $w = 9$. We also display in dashed blue the cardinality of $\mathcal{H}_w$ over the Hamming metric with $t = \lfloor \mathbb{E}[\mathrm{wt}_H(\mathbf{e}) \mid \mathrm{wt}_L(\mathbf{e}) = 7] \rceil$, where t is computed with Equation 11.

As depicted in Figure 5, the cardinality is larger when m is prime. As we can see, the orange plot is close to the dashed blue one. This reveals to us that the statistical bias does not increase by much when changing the metric and that the gap between the orange and blue curve is mostly due to the expected Hamming weight of the error. In fact that statistical decoding over the Lee metric with $\mathrm{wt}_L(\mathbf{e}) = t$ seems as efficient as statistical decoding with the Hamming metric with the error being of weight $\mathbb{E}[\mathrm{wt}_H(\mathbf{e}) \mid \mathrm{wt}_L(\mathbf{e}) = t]$. As explained in Section 3.5, it follows from the fact that the statistical decoding over the Lee metric reduces to a simpler instance of statistical decoding over the Hamming metric where the weight of the error is close to $\mathbb{E}[\mathrm{wt}_H(\mathbf{e})]$ with high probability, as shown with the Chebyshev's inequality in Section 3.5.1.

The following section introduces the ISD techniques for decoding. These algorithms do not scale as well as statistical decoding when changing the metric to the Lee one. More than that, the algorithm performs worse than its Hamming

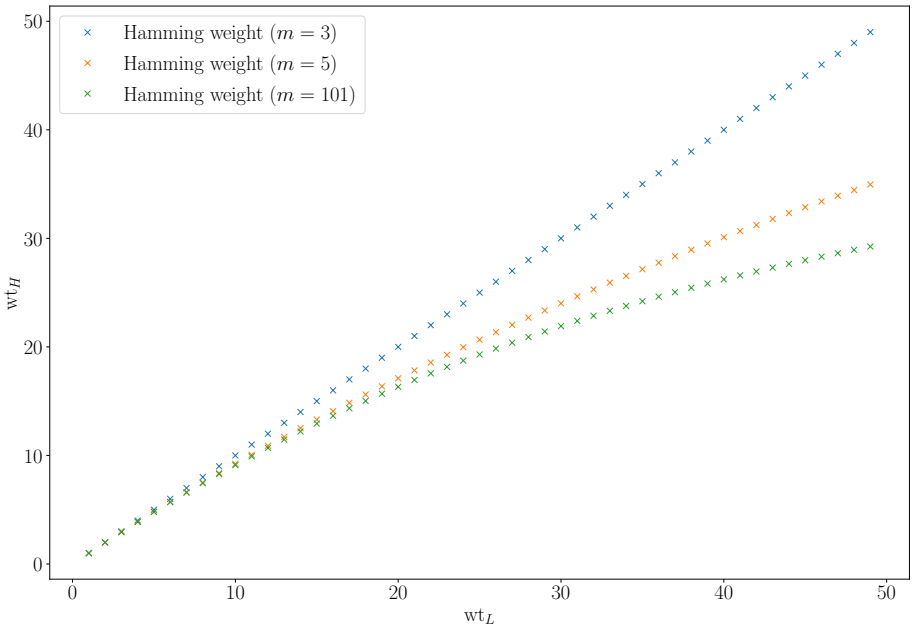


Fig. 8: Expected Hamming weight a uniformly random vector of a fixed Lee weight over $\mathbb{Z}/21\mathbb{Z}$. There are three plots for three different m.

counterpart. This motivates the fact that the Lee metric may lead to smaller public key size PKE than we previously thought. Section 5 shows whether it is necessary to reconsider the public key size of the Lee-McEliece cryptosystem to account for the transformed statistical decoding algorithm.

4 Information Set Decoding

The idea of Information Set Decoding (ISD) was first proposed by Prange [37] in 1962. It has then been improved over time and is now considered the best general decoding algorithm. In this section, we will give an intuitive interpretation of this algorithm, and present the different improvements and their complexity. The main aim of this chapter is not to give all details behind this family of algorithms but to present all the results needed to compare ISD with statistical decoding. For the sake of clarity, we assume without loss of generality that the information set $\mathcal{I}$ is $[\![k]\!]$. Moreover, in the Lee case, we will only consider rings of the form $\mathbb{Z}/p^s\mathbb{Z}$. For the sake of clarity, we will use the same notation for the complexity of these algorithms over the different metrics. There will be no confusion about which metric we are talking about.

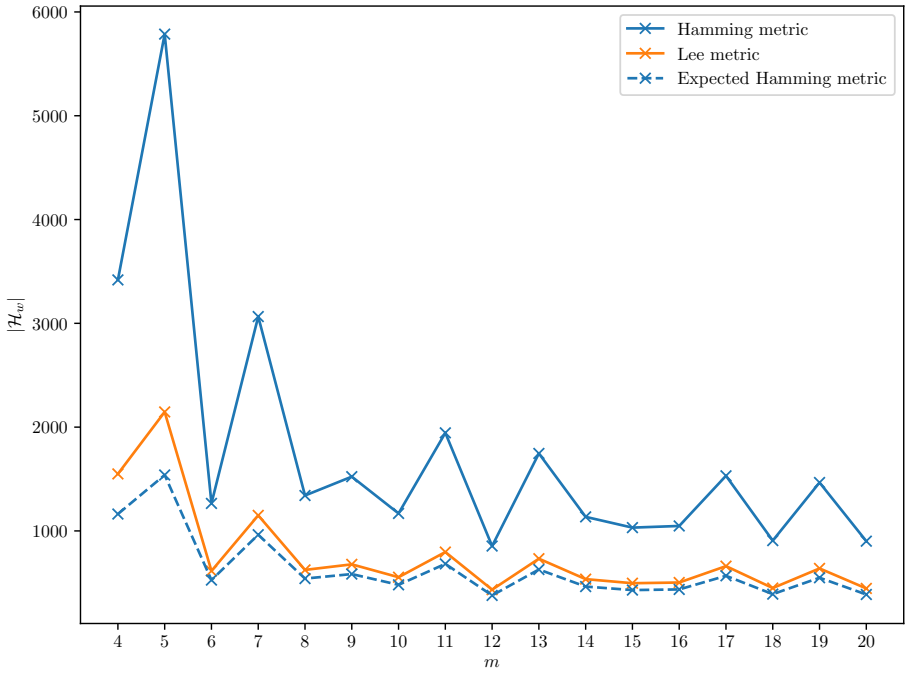


Fig. 9: $|\mathcal{H}_w|$ to achieve 95% probability of success over $(\mathbb{Z}/m\mathbb{Z})$ with $t = 7$, $w = 9$ and $n = 30$ in the Hamming (blue) and Lee (red) metric. The cardinality of $\mathcal{H}_w$ over the Hamming metric with $t = \lfloor \mathbb{E}[\mathrm{wt}_H(\mathbf{e}) \mid \mathrm{wt}_L(\mathbf{e}) = 4] \rceil$ is presented in dashed blue.

4.1 Prange's Original Idea

Let $\mathcal{C} \leq \mathbb{F}_q^n$ be a binary linear code of distance k. The Prange's algorithm [37] requires the following assumption:

Assumption 3 *Let* $\mathbf{e}$ *be the error vector, we assume that* $\mathrm{Supp}(\mathbf{e}) \cap \mathcal{I}(\mathcal{C}) = \emptyset$.

The idea of ISD is to guess the information set of the code and then transform its parity check matrix $\mathbf{H}$ into its systematic form with respect to $\mathcal{I}$ by finding a matrix $\mathbf{U} \in \mathbb{F}_q^{(n-k)\times(n-k)}$ such that $(\mathbf{UH})^{\mathfrak{C}(\bar{\mathcal{I}})} = \mathbf{I}_{n-k}$, where $\bar{\mathcal{I}} \triangleq [\![n]\!] \setminus \mathcal{I}$.

Let's assume that we have found the correct information set $\mathcal{I}$. Observe that by definition of $\mathbf{U}$ and Assumption 3 we have $\mathbf{e}\,(\mathbf{UH})^\top = \mathbf{e}_{\bar{\mathcal{I}}}$. On the other hand, $\mathbf{e}\,(\mathbf{UH})^\top = \mathbf{eH}^\top\mathbf{U}^\top = \mathbf{sU}^\top$, where $\mathbf{s}$ is the syndrome. Note that we can calculate the rightmost expression. Let $\mathbf{s}'$ be $\mathbf{sU}^\top$. As indicated above, if the chosen information set is the correct one, we have $\mathbf{s}' = \mathbf{e}_{\bar{\mathcal{I}}}$. A way of checking that our guess is indeed the information set is to test whether the Hamming weight of $\mathbf{s}'$ is t as expected. This is essentially how the Prange algorithm works.

For the sake of simplicity, we will use the following notation for sets of indices.

$$\mathcal{B}_{k,n} \triangleq S^n_{2,H}(\mathbf{0}, k).$$

The algorithm for any finite field $\mathbb{F}_q$ embedded with the Hamming metric is described in Algorithm 1.

Algorithm 1 Prange's Algorithm over $\mathbb{F}_q$.

input $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}, \mathbf{s} \in \mathbb{F}_q^{n-k}, t \in \mathbb{N}$
$\mathbf{s}' \leftarrow \mathbf{0}_{n-k}$
while $\mathrm{wt}_H(\mathbf{s}') \neq t$ **do**
  // Guess the information set
  $I \leftarrow\!\!\$\ \mathcal{B}_{k,n}$
  $\mathbf{U} \leftarrow \mathbb{F}_q^{(n-k)\times(n-k)}$ such that $(\mathbf{UH})^{\mathfrak{C}(\bar{I})} = \mathbf{I}_{n-k}$
  $\mathbf{s}' \leftarrow \mathbf{s}\mathbf{U}^\top$
endwhile
return $\mathbf{e}$ such that $\mathbf{e}_I = \mathbf{0}_K \wedge \mathbf{e}_{\bar{I}} = \mathbf{s}'$

The average running time of this algorithm is given in Violetta Weger's thesis, it is given here for the sake of completeness.

Theorem 13 (Complexity of Prange's Algorithm [45, Theorem 4.1.1]). *The running time of Prange's algorithm over $\mathbb{F}_q$ is given by*

$$T_{\text{PRANGE}} = \binom{n-k}{t}^{-1}\binom{n}{t}(n-k)^2(n+1)(\lceil \log_2 q\rceil + \lceil \log_2 q\rceil^2).$$

We now translate this algorithm into the Lee metric. The only difference is in the form of the parity check matrix. However instead of considering the full systematic form

$$\begin{pmatrix} \mathbf{B_{1,1}} & \mathbf{B_{1,2}} & \cdots & \mathbf{B_{1,m-1}} & \mathbf{B_{1,m}} & \mathbf{I_{n-K}} \\ p\mathbf{B_{2,1}} & p\mathbf{B_{2,2}} & \cdots & p\mathbf{B_{2,m-1}} & p\mathbf{I_{k_m}} & 0 \\ p^2\mathbf{B_{3,1}} & p^2\mathbf{B_{3,2}} & \cdots & p^2\mathbf{I_{k_{m-1}}} & \mathbf{0} & 0 \\ \vdots & \vdots & & \vdots & \vdots & \vdots \\ p^{m-1}\mathbf{B_{m,1}} & p^{m-1}\mathbf{I_{k_2}} & \cdots & \mathbf{0} & \mathbf{0} & 0 \end{pmatrix},$$

we will use the following simplified form:

$$\begin{pmatrix} \mathbf{A} & \mathbf{I_{n-K}} \\ p\mathbf{B} & 0 \end{pmatrix}, \tag{13}$$

with $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K)\times K}$ and $\mathbf{B} \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{(K-k_1)\times K}$.

Let s be a strictly positive integer. The general algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ embedded with the Lee metric is presented in Algorithm 2. This algorithm was first designed by Violetta Weger *et al.* in [46]. We can retrieve the algorithm for the special field case $\mathbb{Z}/p\mathbb{Z}$ by substituting s by 1.

Algorithm 2 Prange's Algorithm (Lee) over $\mathbb{Z}/p^s\mathbb{Z}$.

input $\mathbf{H} \in (\mathbb{Z}/p\mathbb{Z})^{(n-k_1)\times n}, \mathbf{s} \in (\mathbb{Z}/p\mathbb{Z})^{n-k_1}, t \in \mathbb{N}$
$\mathbf{s_1} \leftarrow \mathbf{0}_{n-K}$
$\mathbf{s_2} \leftarrow \mathbf{0}_{K-k_1}$
while $\mathrm{wt}_L(\mathbf{s_1}) \neq t \vee \mathrm{wt}_L(\mathbf{s_2}) \neq 0$ **do**
  // Guess the information set
  $I \xleftarrow{\$} \mathcal{B}_{K,n}$
  $\mathbf{U} \leftarrow (\mathbb{Z}/p\mathbb{Z})_q^{(n-k_1)\times(n-k_1)}$ such that $(\mathbf{UH})^{\mathfrak{C}(\bar{\mathcal{I}})} = \begin{pmatrix} \mathbf{I}_{n-K} \\ \mathbf{0}_{(K-k_1)\times(n-K)} \end{pmatrix}$
  and $(\mathbf{UH})^{\mathfrak{C}(\mathcal{I})} = \begin{pmatrix} \mathbf{A} \\ p\mathbf{B} \end{pmatrix}$ with $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K)\times K}$
  and $\mathbf{B} \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{(K-k_1)\times(n-K)}$
  $[\mathbf{s_1}, p\mathbf{s_2}] \leftarrow \mathbf{sU}^\top$
endwhile
return $\mathbf{e}$ such that $\mathbf{e}_I = 0 \wedge \mathbf{e}_{\bar{I}} = \mathbf{s}'$

The average running is given in Violetta Weger's thesis.

Theorem 14 (Complexity of Prange's Algorithm in the Lee metric [45, Theorem 5.3.1]). *The Prange's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric has complexity*

$$T_{\text{PRANGE}} = F(n-K,t,p^s)^{-1}F(n,t,p^s)(n-k_1)^2(n+1)\left(\lceil\log_2(p^s)\rceil + \lceil\log_2(p^s)\rceil^2\right),$$

where F is defined in Equation 1.

4.2 Lee-Brickell's Algorithm

The idea behind Lee-Brickell's improvement is to reduce the number of iterations while increasing the cost of one iteration [23]. It allows some erroneous positions to be in our guessed information set. More precisely, we allow v errors to lie inside our information set and $t - v$ outside. The variable v is a parameter given as input to the algorithm. Once we have transformed $\mathbf{H}$ into its systematic form, we have to iterate over all error vectors of weight v in order to find the correct error

vector. In other words, Lee-Brickell's improvement is a compromise between the number of iterations and the time list executing the new inner loop.

The algorithm was first introduced by Lee and Brickell in [23] and then generalized by Peters over any finite field in [36]. Algorithm 3 depicts the pseudocode of the algorithm over finite fields. Recall that $S^k_{q,H}(\mathbf{0}, v)$ denotes the set of vector of Hamming weight q (Definition 3).

Algorithm 3 Lee-Brickell's Algorithm over $\mathbb{F}_q$.

input $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}, \mathbf{s} \in \mathbb{F}_q^{n-k}, t \in \mathbb{N}, v < \min\{t, k\}$
// Guess the information set
$I \leftarrow\$ \mathcal{B}_{k,n}$
$\mathbf{U} \leftarrow \mathbb{F}_q^{(n-k)\times(n-k)}$ such that $(\mathbf{UH})^{\mathfrak{C}(\bar{\mathcal{I}})} = \mathbf{I}_{n-k}$
$\mathbf{A} \leftarrow (\mathbf{UH})^{\mathfrak{C}(\mathcal{I})} \in \mathbb{F}_q^{(n-k)\times k}$
$\mathbf{s}' \leftarrow \mathbf{sU}^\top$
// Guess the v error positions.
foreach $\mathbf{e}_I \in S^k_{q,H}(\mathbf{0}, v)$ **do**
  if $\mathrm{wt}_H(\mathbf{s}' - \mathbf{e}_I\mathbf{A}^\top) = t - v$ **then**
    return $\tilde{\mathbf{e}}$ such that $\tilde{\mathbf{e}}_I = \mathbf{e}_I \wedge \tilde{\mathbf{e}}_{\bar{I}} = \mathbf{s}' - \mathbf{e_I}\mathbf{A}^\top$
  fi
endforeach
goto 1

Its complexity is given in [45], we state the result in the following theorem:

Theorem 15 (Complexity of Lee-Brickell's Algorithm [45, Theorem 4.2.1]). *The complexity of the Lee-Brickell's Algorithm over $\mathbb{F}_q$ is*

$$T_{\text{LEE-BRICKELL}} = \binom{k}{v}^{-1}\binom{n-k}{t-v}^{-1}\binom{n}{t}\left((n-k)^2(n+1) + \binom{k}{v}(q-1)^v \cdot \min\{n-k, \frac{q}{q-1}(t-v+1)\}v\right)\left(\lceil \log_2 q\rceil + \lceil \log_2 q\rceil^2\right).$$

The algorithm naturally extends over $\mathbb{Z}/p^s\mathbb{Z}$ in the Lee metric. This extension was first introduced in [47]. It is presented in Algorithm 4.

An estimate of the complexity is done in [45]. We state the result below.

Theorem 16 (Complexity of Lee-Brickell's Algorithm in the Lee Metric [45, Theorem 5.3.3]). *The complexity of the Lee-Brickell's algorithm over*

Algorithm 4 Lee-Brickell's Algorithm (Lee) over $\mathbb{Z}/p^s\mathbb{Z}$.

input $\mathbf{H} \in (\mathbb{Z}/p\mathbb{Z})^{(n-k_1)\times n}, \mathbf{s} \in (\mathbb{Z}/p\mathbb{Z})^{n-k_1}, t \in \mathbb{N}$
// Guess the information set
$I \leftarrow\$ \mathcal{B}_{K,n}$
$\mathbf{U} \leftarrow (\mathbb{Z}/p\mathbb{Z})_q^{(n-k_1)\times(n-k_1)}$ such that $(\mathbf{UH})^{\mathfrak{C}(\bar{\mathcal{I}})} = \begin{pmatrix} \mathbf{I}_{n-K} \\ \mathbf{0}_{(K-k_1)\times(n-K)} \end{pmatrix}$
and $(\mathbf{UH})^{\mathfrak{C}(\mathcal{I})} = \begin{pmatrix} \mathbf{A} \\ p\mathbf{B} \end{pmatrix}$ with $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K)\times K}$
and $\mathbf{B} \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{(K-k_1)\times(n-K)}$
$[\mathbf{s}_1, p\mathbf{s}_2] \leftarrow \mathbf{sU}^\top$ where $\mathbf{s}_1 \in (\mathbb{Z}/p^s\mathbb{Z})^{n-K}$ and $\mathbf{s}_2 \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{K-k_1}$
foreach $\mathbf{e}_I \in S^K_{p^s,H}(\mathbf{0}, v)$ **do**
  if $\mathbf{e}_I\mathbf{B}^\top = \mathbf{s}_2$ **then**
    if $\mathrm{wt}_L(\mathbf{s}' - \mathbf{e}_I\mathbf{A}^\top) = t - v$ **then**
      return $\tilde{\mathbf{e}}$ such that $\tilde{\mathbf{e}}_I = \mathbf{e}_I \wedge \tilde{\mathbf{e}}_{\bar{I}} = \mathbf{s}' - \mathbf{e_I}\mathbf{A}^\top$
    fi
  fi
endforeach
goto 1

$\mathbb{Z}/p^s\mathbb{Z}$ embedded with the Lee metric is

$$
\begin{aligned}
&T_{\text{LEE-BRICKELL}} = \\
&F(K, v, p^s)^{-1} F(n-K, t-v, p^s)^{-1} F(n, t, p^s) \Bigg((n-k_1)^2 (n+1) \\
&+ F(K, v, p^s) \min\{v, K\} (K-k_1) \\
&+ F(K, v, p^s) \min\left\{\mu_{p^s}^{-1}(t-v+1), n-K\right\} \min\{v, K\} \Bigg) \\
&\cdot \left(\lceil \log_2(p^s) \rceil + \lceil \log_2(p^s) \rceil^2 \right).
\end{aligned}
$$

4.3 Stern's Algorithm

Stern's algorithm is another improvement of Prange's idea. It has first been presented in [43]. It combines the ideas of Prange's algorithm with the famous Birthday paradox and a *zero-window*. The latter was first introduced by Leon in [25].

The zero-window is a set of indices, denoted by Z, that lies outside the information set, where no errors happen. We consider I to be the information

set, Z the zero-window of size ℓ, and $J = \overline{I \cup Z}$. For the sake of clarity, we assume that $Z = [\![\ell]\!] + k$. The algorithm allows $2v$ errors in I. If we are lucky, the error vector $\mathbf{e}$ has weight $2v$ when restricted to I, 0 when restricted to Z, and $t - 2v$ otherwise. With our specific choice of I and Z, we have $\mathbf{e} = [\mathbf{e}_I, \mathbf{0}_\ell, \mathbf{e}_J]$. Let $\mathbf{U}$ be the matrix that transform $\mathbf{H}$ into its systematic form with $\mathbf{A} \in \mathbb{F}_q^{\ell \times k}$ and $\mathbf{B} \in \mathbb{F}_q^{(n-k-\ell)\times k}$. We have that

$$(\mathbf{UH})^\top = \begin{pmatrix} \mathbf{A}^\top & \mathbf{B}^\top \\ \mathbf{I}_\ell & \mathbf{0}_{\ell\times(n-k-\ell)} \\ \mathbf{0}_{(n-k-\ell)\times\ell} & \mathbf{I}_{n-k-\ell} \end{pmatrix}.$$

On one hand we have $\mathbf{e}(\mathbf{UH})^\top = [\mathbf{s}_1, \mathbf{s}_2] = \mathbf{sU}^\top$, with $\mathbf{s}_1 \in \mathbb{F}_q^\ell$ and $\mathbf{s}_2 \in \mathbb{F}_q^{n-k-\ell}$. On the other hand, we have that $\mathbf{e}(\mathbf{UH})^\top = [\mathbf{s}_1, \mathbf{s}_2] = [\mathbf{e}_I\mathbf{A}^\top, \mathbf{e}_I\mathbf{B}^\top + \mathbf{e}_J]$. We can deduce the two followings conditions:

$$\mathbf{e}_I\mathbf{A}^\top = \mathbf{s}_1 \tag{14}$$

$$\mathbf{e}_I\mathbf{B}^\top + \mathbf{e}_J = \mathbf{s}_2. \tag{15}$$

We first seek to find $\mathbf{e}_I$ using the Birthday paradox under the condition given by Equation 14 and then use Equation 15 to find $\mathbf{e}_j$ and check that it has Hamming weight $t - 2v$.

To find $\mathbf{e}_I$, we proceed as follows: We partition I into two sets, X and Y. The idea is to partition $\mathbf{e}_I$ into $\mathbf{e}_X$ and $\mathbf{e}_Y$ such that $\mathrm{wt}_H(\mathbf{e}_X) = \mathrm{wt}_H(\mathbf{e}_Y) = v$. Looking at Equation 14, we obtain a new condition

$$\begin{aligned} &\mathbf{e}_I\mathbf{A}^\top = \sigma_X(\mathbf{e}_X)\mathbf{A}^\top + \sigma_X(\mathbf{e}_Y)\mathbf{A}^\top = \mathbf{s}_q \\ \Leftrightarrow\ &\sigma_X(\mathbf{e}_X) = \mathbf{s}_1 - \sigma_Y(\mathbf{e}_Y)\mathbf{A}^\top, \end{aligned} \tag{16}$$

where $\sigma_\cdot$ is the cannonical projection defined in Section 2.1.

Finally, we compute the set of vectors of the form of the left-hand side of Equation 16 and the set corresponding to the right-hand side. If a collision occurs, we have found a candidate for $\mathbf{e}_I$. Then as explained before, we can use Equation 15 to compute $\mathbf{e}_J$ and check that the Hamming weight is the expected one. We now give the entire algorithm over $\mathbb{F}_q$. The generalization to arbitrary finite fields was done by Peters in [36], it is presented in Algorithm 5

The complexity of the algorithms is given in Violetta Weger's thesis.

Theorem 17 (Complexity of Stern's Algorithm [45, Theorem 4.3.1]). *The complexity of Stern's algorithm over $\mathbb{F}_q$ embedded with the Hamming metric*

Algorithm 5 Stern's Algorithm over $\mathbb{F}_q$.

input $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}, \mathbf{s} \in \mathbb{F}_q^{n-k}, t \in \mathbb{N}, k = m_1 + m_2, \ell < n-k,$
$v < \min\{m_1, m_2, \lfloor \frac{t}{2} \rfloor\}$
// Guess the information set
$I \leftarrow\!\$\ \mathcal{B}_{k,n}$
// Choose a zero-window
$Z \subset \bar{I}$ such that $|Z| = \ell$
$J \leftarrow \overline{Z \cup I}$
$X, Y \leftarrow$ partitions of I of respective sizes m_1 and m_2
$\mathbf{U} \leftarrow \mathbb{F}_q^{(n-k)\times(n-k)}$ such that $(\mathbf{UH})^{\mathfrak{C}(\mathcal{I})} = \begin{pmatrix}\mathbf{A}\\ \mathbf{B}\end{pmatrix}$,
$(\mathbf{UH})^{\mathfrak{C}(\mathcal{Z})} = \begin{pmatrix}\mathbf{I}_l\\ \mathbf{0}_{(n-k-\ell)\times\ell}\end{pmatrix}$ and $(\mathbf{UH})^{\mathfrak{C}(\mathcal{J})} = \begin{pmatrix}\mathbf{0}_{l\times(n-k-\ell)}\\ \mathbf{I}_{n-k-\ell}\end{pmatrix}$
where $\mathbf{A} \in \mathbb{F}_q^{l\times k}$ and $\mathbf{B} \in \mathbb{F}_q^{(n-k-\ell)\times k}$
$[\mathbf{s}_1, \mathbf{s}_2] \leftarrow \mathbf{sU}^\top$, where $\mathbf{s}_1 \in \mathbb{F}_q^\ell$ and $\mathbf{s}_2 \in \mathbb{F}_q^{n-k-\ell}$
// Partitions.
$\mathcal{L}_1 \leftarrow \{(\mathbf{e}_X, \sigma_X(\mathbf{e}_X)\mathbf{A}^\top) \mid \mathbf{e}_X \in \mathbb{F}_q^{m_1} \wedge \mathrm{wt}_H(\mathbf{e}_X) = v\}$
$\mathcal{L}_2 \leftarrow \{(\mathbf{e}_Y, \mathbf{s}_1 - \sigma_Y(\mathbf{e}_Y)\mathbf{A}^\top) \mid \mathbf{e}_Y \in \mathbb{F}_q^{m_2} \wedge \mathrm{wt}_H(\mathbf{e}_Y) = v\}$
// Search for collisions.
foreach $(\mathbf{e}_X, \mathbf{a}) \in \mathcal{L}_1$ **do**
  foreach $(\mathbf{e}_Y, \mathbf{a}) \in \mathcal{L}_2$ **do**
    if $\mathrm{wt}_H(\mathbf{s}_2 - (\sigma_X(\mathbf{e}_X) + \sigma_Y(\mathbf{e}_Y))\mathbf{B}^\top) = t - 2v$ **then**
      return $\tilde{\mathbf{e}}$ such that $\tilde{\mathbf{e}}_I = \sigma_X(\mathbf{e}_X) + \sigma_Y(\mathbf{e}_Y) \wedge \tilde{\mathbf{e}}_Z = \mathbf{0}_l$
      $\wedge\ \tilde{\mathbf{e}}_J = \mathbf{s}_2 - (\sigma_X(\mathbf{e}_X) + \sigma_Y(\mathbf{e}_Y))\mathbf{B}^\top$
    fi
  endforeach
endforeach
goto 1

is

$$
\begin{aligned}
&T_{\text{STERN}} = \\
&\binom{m_1}{v}^{-1}\binom{m_2}{v}^{-1}\binom{n-k-\ell}{t-2v}^{-1}\binom{n}{t} \\
&\cdot\left((n-k)^2(n+1)\left(\lceil\log_2(q)\rceil + \lceil\log_2(q)\rceil^2\right) + (m_1+m_2)\,\ell\,\lceil\log_2(q)\rceil^2\right. \\
&+\ell\left(L_q(m_1, v) + L_q(m_2, v) + \binom{m_2}{v}(q-1)^v\right)\lceil\log_2(q)\rceil \\
&+\frac{\binom{m_1}{v}\binom{m_2}{v}(q-1)^{2v}}{q^\ell}\min\left\{n-k-\ell, \frac{q}{q-1}(t-2v+1)\right\} \\
&\left.\cdot 2v\left(\lceil\log_2(q)\rceil^2 + \lceil\log_2(q)\rceil\right)\right).
\end{aligned}
$$

This algorithm has been extended to the finite ring $\mathbb{Z}/p^s\mathbb{Z}$ equipped with the Lee metric in [46]. We describe the pseudocode in Algorithm 6.

Algorithm 6 Stern's Algorithm (Lee) over $\mathbb{Z}/p^s\mathbb{Z}$.

input $\mathbf{H} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-k_1)\times n}, \mathbf{s} \in (\mathbb{Z}/p^s\mathbb{Z})^{n-k_1}, t \in \mathbb{N}, K = m_1 + m_2,$
$\ell < n - K, v < \min\{\lfloor \frac{p^s-1}{2} \rfloor m_1, \lfloor \frac{p^s-1}{2} \rfloor m_2, \lfloor \frac{t}{2} \rfloor\}$
// Guess the information set
$I \leftarrow\!\!\$\ \mathcal{B}_{K,n}$
// Choose a zero-window
$Z \subset \bar{I}$ such that $|Z| = \ell$
$J \leftarrow \overline{Z \cup I}$
$X, Y \leftarrow$ partitions of I of respective sizes m_1 and m_2
$\mathbf{U} \leftarrow (\mathbb{Z}/p^s\mathbb{Z})^{(n-k_1)\times(n-k_1)}$ such that $(\mathbf{UH})^{\mathfrak{C}(\mathcal{I})} = \begin{pmatrix} \mathbf{A} \\ \mathbf{B} \\ p\mathbf{C} \end{pmatrix}$,
$(\mathbf{UH})^{\mathfrak{C}(\mathcal{Z})} = \begin{pmatrix} \mathbf{I}_\ell \\ \mathbf{0}_{(n-K-\ell)\times\ell} \\ \mathbf{0}_{(K-k_1)\times\ell} \end{pmatrix}$ and $(\mathbf{UH})^{\mathfrak{C}(\mathcal{J})} = \begin{pmatrix} \mathbf{0}_{\ell\times(n-K-\ell)} \\ \mathbf{I}_{n-K-\ell} \\ \mathbf{0}_{(K-k_1)\times(n-K-\ell)} \end{pmatrix}$
where $\mathbf{A} \in (\mathbb{Z}/p^s\mathbb{Z})^{\ell\times K}, \mathbf{B} \in (\mathbb{Z}/p^s\mathbb{Z})^{(n-K-\ell)\times K}$ and $\mathbf{C} \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{K-k_1}$
$[\mathbf{s}_1, \mathbf{s}_2, p\mathbf{s}_3] \leftarrow \mathbf{sU}^\top$, where $\mathbf{s}_1 \in (\mathbb{Z}/p^s\mathbb{Z})^\ell, \mathbf{s}_2 \in (\mathbb{Z}/p^s\mathbb{Z})^{n-K-\ell}$ and $\mathbf{s}_3 \in (\mathbb{Z}/p^{s-1}\mathbb{Z})^{K-k_1}$
// Partitions.
$\mathcal{L}_1 \leftarrow \{(\mathbf{e}_X, \sigma_X(\mathbf{e}_X)\mathbf{A}^\top, \sigma_X(\mathbf{e}_X)\mathbf{C}^\top) \mid \mathbf{e}_X \in (\mathbb{Z}/p^s\mathbb{Z})^{m_1} \wedge \mathrm{wt}_H(\mathbf{e}_X) = v\}$
$\mathcal{L}_2 \leftarrow \{(\mathbf{e}_Y, \mathbf{s}_1 - \sigma_Y(\mathbf{e}_Y)\mathbf{A}^\top, \mathbf{s}_3 - \sigma_Y(\mathbf{e}_Y)\mathbf{C}^\top) \mid \mathbf{e}_Y \in (\mathbb{Z}/p^s\mathbb{Z})^{m_2} \wedge \mathrm{wt}_H(\mathbf{e}_Y) = v\}$
// Search for collisions.
foreach $(\mathbf{e}_X, \mathbf{a}, \mathbf{b}) \in \mathcal{L}_1$ **do**
  foreach $(\mathbf{e}_Y, \mathbf{a}, \mathbf{b}) \in \mathcal{L}_2$ **do**
    if $\mathrm{wt}_H(\mathbf{s}_2 - (\sigma_X(\mathbf{e}_X) + \sigma_Y(\mathbf{e}_Y))\mathbf{B}^\top) = t - 2v$ **then**
      return $\tilde{\mathbf{e}}$ such that $\tilde{\mathbf{e}}_I = \sigma_X(\mathbf{e}_X) + \sigma_Y(\mathbf{e}_Y) \wedge \tilde{\mathbf{e}}_Z = \mathbf{0}_\ell$
      $\wedge\, \tilde{\mathbf{e}}_J = \mathbf{s}_2 - (\sigma_X(\mathbf{e}_X) + \sigma_Y(\mathbf{e}_Y))\mathbf{B}^\top$
    fi
  endforeach
endforeach
goto 1

Before giving an expression for the complexity of this algorithm, we need the following result:

Proposition 22 (Expected Lee Weight [39, Problem 10.15]). *Let x be randomly drawn in $\mathbb{Z}/p^s\mathbb{Z}$. Its expected Lee weight, written as μ_{p^s}, is*

$$\mu_{p^s} = \begin{cases} \frac{p^s}{4} & \text{if } p = 2, \\ \frac{p^{2s}-1}{4p^s} & \text{otherwise.} \end{cases}$$

As with the other algorithms, the average complexity of this algorithm is given in Violetta Weger's thesis.

Theorem 18 (Complexity of Stern's Algorithm in the Lee Metric [45, Theorem 5.3.5]). *The complexity of the Lee-Brickell's algorithm over $\mathbb{Z}/p^s\mathbb{Z}$ embedded with the Lee metric is*

$$\begin{aligned} &T_{\text{STERN}} = \\ &F(m_1, v, p^s)^{-1} F(m_2, v, p^s)^{-1} F(n-K-\ell, t-2v, p^s)^{-1} F(n, t, p^s) \\ &\cdot \left(\lceil \log_2(p^s) \rceil + \lceil \log_2(p^s) \rceil^2\right) \Bigg((n-k_1)^2 (n+1) \\ &+ (K - k_1 + \ell) \left(F\left(m_1, v, p^s\right) \min\left\{v, m_1\right\} + F\left(m_2, v, p^s\right) \min\left\{v, m_2\right\}\right) \\ &+ \frac{F\left(m_1, v, p^s\right) F\left(m_2, v, p^s\right)}{(p^s)^{\ell+K-k_1}} \min\left\{\mu_{p^s}^{-1}(t-2v+1), n-K-\ell\right\} \min\{2v, K\} \Bigg). \end{aligned}$$

4.4 Becker-Joux-May-Meurer's Improvement

BJMM algorithm [4] is an improvement of Stern's algorithm. It uses a combination of Wagner's idea [44] and *representation techniques*. Fix an integer strictly positive integer a, Wagner's approach is to subdivide the searched error vector into 2^a subvectors and store them in a list with their respective syndromes. Representation allows vector in $\mathcal{L}_1$ and $\mathcal{L}_2$ having weight $\frac{v}{2} + \epsilon$ where ϵ is a positive integer that represents the overlapping part that should be canceled when adding vector of the two lists. The optimal choice of a depends on the metric [47]: $a = 3$ for Hamming and $a = 2$ for Lee.

The generalization for $\mathbb{Z}/m\mathbb{Z}$ embedded with the Lee metric was done by Violetta Weger *et. al* in [47]. We will not state the complexity result here but note that they give a link to the `SAGE` source code that allows them to compute the workfactor of BJMM over the Lee metric while optimizing all parameters.

BJMM algorithm is the best improvement of Prange's algorithm that we will consider when comparing ISD algorithms with statistical decoding. The MMT algorithm [27] is a similar algorithm where vectors have no overlap, however, non-asymptotic analysis of this algorithm over the binary field shows that it is less efficient than BJMM [15]. We now have all the results needed in order to compare ISD with the statistical decoding algorithm over the Lee metric.

5 Comparing Information Set Decoding with Statistical Decoding

This section compares the two families of algorithms over different rings and the two metrics. As we will see, it is not true that ISD algorithms are always faster than statistical decoding. The performance gap between them reduces when the structure size increases. As a result, statistical decoding outperforms ISD algorithms for rings large enough.

For computational efficiency, we will consider the ring $\mathbb{Z}/2^s\mathbb{Z}$ when focusing on the general ring case. Indeed, as we have seen in Section 3.4.4, we can implement a much faster function for $\widetilde{\mathfrak{I}}$ when working over $\mathbb{Z}/2^s\mathbb{Z}$. To compare statistical decoding and ISD algorithms we need to know the complexity of generating the set $\mathcal{H}_w$. As in [31], we propose a naive sample and reject algorithm and derive a rough approximation of its complexity. The algorithm is generic: it does not depend on any particular version of statistical decoding. We recall that by Definition 3, $S^n_{q,H}(\mathbf{0}, w)$ is the set of vectors in $\mathbb{F}_q^n$ of Hamming weight w.The pseudocode is described in Algorithm 7 over $\mathbb{F}_q$ but it adapts naturally to $\mathbb{Z}/m\mathbb{Z}$.

Algorithm 7 Generation of $\mathcal{H}_w$.

```
1 : input H ∈ F_q^{(n−k)×n}, N ∈ ℕ
2 : H_w ← ∅
3 : while |H_w| ≠ N do
4 :     h ←$ S^n_{q,H}(0, w)
5 :     if hH^⊤ ≠ 0
6 :         H_w ← H_w ∪ {h}
7 :     fi
8 : endwhile
9 : return H_w
```

If we assume that the predicate of the algorithm has a fifty percent chance of being correct, then a rough approximation of its complexity is given by the following theorem:

Theorem 19 (Complexity of Generating $\mathcal{H}_w$). *The complexity of the sample and reject algorithm over $\mathbb{F}_q^n$, denoted by $T_{\text{GEN}}(N)$, is roughly approximated by*

$$T_{\text{GEN}}(N) = \frac{2n(n-k)N}{0.95R},$$

where 0.95 represents the probability of success of the algorithm and R is the rate of the code.

Proof. We assume that drawing a vector at random within $S^n_{q,H}(\mathbf{0}, w)$ takes constant time. Using the naive matrix multiplication algorithm, matrix multiplication needs $n(n-k)$ q-ary operations. Assuming that elements of the same Hamming weight are well distributed within the code $\mathcal{C}$ and its dual $\mathcal{C}^\perp$ when it follows that we need in expectation $\frac{1}{R}$ new vectors to get admissible one. Finally, $\frac{N}{0.95}$ accounts for the fact that our algorithm succeeds with probability 95% and that we need to append N distinct elements in the set $\mathcal{H}_w$. □

Now we can give an expression for the running time of the statistical decoding algorithm without any precomputation.

Theorem 20 (Complexity of Statistical Decoding Without any Precomputation). *Let N be the cardinality of $\mathcal{H}_w$. We denote by $T_{\text{SD}}(N)$ the complexity of statistical decoding without precomputation. It can be expressed as:*

$$T_{\text{SD}}(N) = T_{\text{GEN}}(N) + T'_{\text{SD}}(N).$$

Proof. The total time complexity is just the complexity of the precomputation and of the algorithm itself. □

We are interested in comparing the so-called *workfactor* of these algorithms. The workfactor is defined to be the base two logarithm of the number of m-ary operations, that is the complexity of the algorithm. Note that we are comparing algorithms over structures of the same size; thus, the comparisons make sense.

Stern and BJMM algorithms are chosen since they are the best improvement of the classical Prange's we have seen. The methodology and the parameters are set as in Violetta Weger's thesis [45] so we can retrieve her results and compare them to our results. We fix the rate of the minimum distance d so that it satisfies the Gilbert-Varshamov bound in the current metric. The bound is motivated by the fact that random codes achieve this bound with high probability. We let t be as large as possible, *i.e.*, $t = \lfloor \frac{d-1}{2} \rfloor$. Whenever we work with integer residue rings, we will only consider rings of the type $\mathbb{Z}/p^s\mathbb{Z}$. We will assume that the subtype of the code is (k_1, k_2), so we can use the simplified form of the parity check matrix introduced in Equation 13. Finally, we arbitrarily choose $w = \frac{n}{3}$ for the statistical decoding algorithm.

A link to the `SAGE` source code is given in her thesis[2]. This allows us to quickly implement the function that calculates the workfactors of the ISD variants. These functions also implement the optimization of the three additional parameters of Stern's improvement. The restriction over rings of the form $\mathbb{Z}/2^s\mathbb{Z}$ allows us to use the simplified expression for $\mathfrak{I}$ presented in Section 3.4.4. This improvement decreases the number of recursive calls from quadratic to log-linear.

As mentioned in Violetta Weger's thesis, there are multiple choices of k_1 over $\mathbb{Z}/p^s\mathbb{Z}$ and the optimal choice is either the largest possible value or the smallest one. To account for this, each time we plot the workfactor of Stern's algorithm we use a dashed green line when k_1 is as small as possible and a green line

[2] It seems that the implementation of the function `countgivensupp` misses the `floor` function in one of its summation bound.

otherwise. The workfactor of statistical decoding is shown in red while that of its precomputation is shown in blue. The sum of these two workfactors gives the workfactor of the algorithm without taking into account any precomputation, this total workfactor is colored in purple.

Figure 10 presents a comparison of the different workfactors over the binary fields $(\mathbb{Z}/2\mathbb{Z})^n$. In this figure and more generally, in the following ones, the abscissa represents n while the ordinate depicts the workfactor. As we would expect from Stern and BJMM algorithms, they perform better than statistical decoding. Note that whenever $n \gtrapprox 50$, even with the help of precomputation, statistical decoding requires more binary operations than its ISD counterpart.

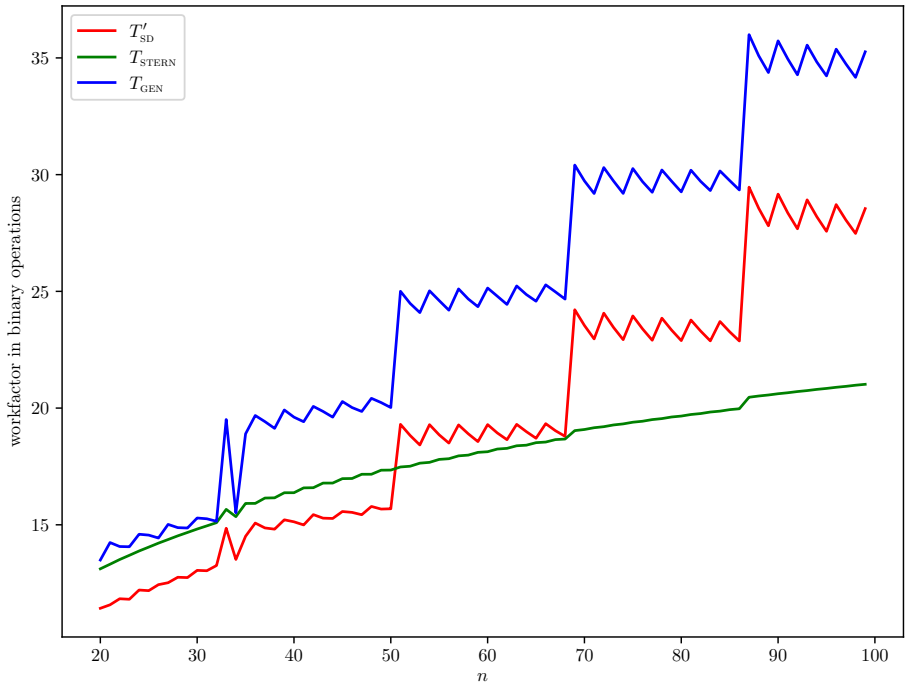


Fig. 10: Workfactors for statistical decoding and Stern algorithms over $(\mathbb{Z}/2\mathbb{Z})^n$.

We now seek to compare these algorithms on a larger field over the Lee metric. Robert Niebuhr showed in [31] that over $\mathbb{F}_q$ embedded with the Hamming metric if q is sufficiently large, then statistical decoding outperforms Stern's algorithm. We would expect this observation to hold over the Lee metric since we have shown in Section 3 that statistical decoding in the Lee metric can only perform better than statistical decoding in the Hamming metric. Figure 11 and Figure 12 respectively depict the workfactor of the algorithms over $\mathbb{Z}/2^8\mathbb{Z}$ and $\mathbb{Z}/2^9\mathbb{Z}$.

Looking at the gap between statistical decoding plots and the other, we can conclude that Niebuhr's observation is still correct over the Lee metric.

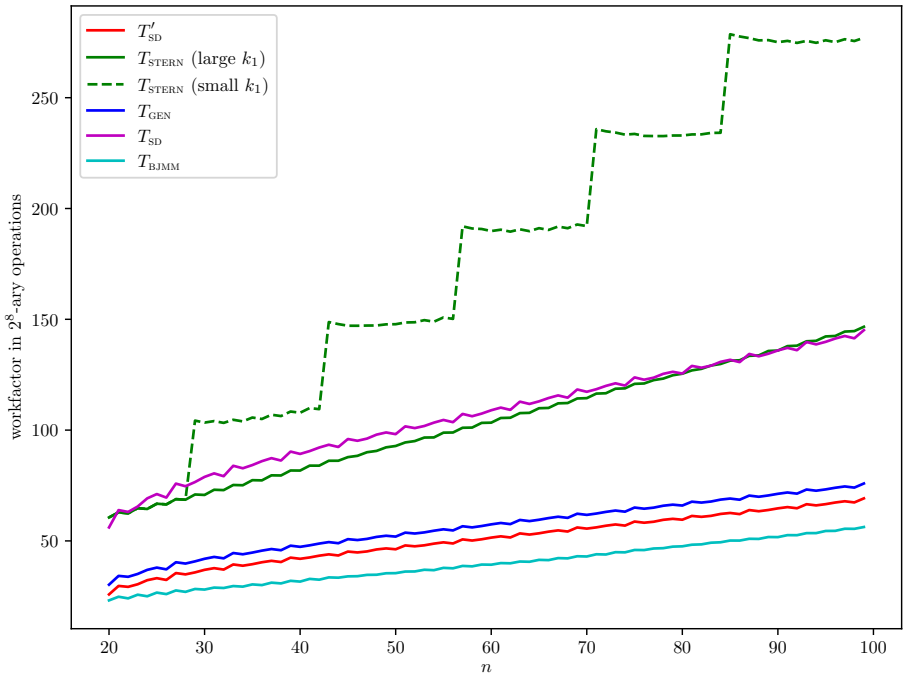


Fig. 11: Workfactors for statistical decoding and Stern algorithms over $(\mathbb{Z}/2^8\mathbb{Z})^n$ embedded with the Lee metric.

We see that doubling the ring size reduces the total workfactor of statistical decoding. ISD algorithms are more affected when increasing the ring size than statistical decoding. In particular, over the larger ring, it appears that statistical without precomputation is more efficient than Stern's algorithm for any $n \lessapprox 60$. This is consistent with Niebuhr's observation. However, even over a ring of size 2^9, BJMM algorithm is still more efficient than statistical decoding over the Lee metric.

Looking at the workfactor of statistical decoding without its precomputation, we see that for $n \lessapprox 30$ over $\mathbb{Z}/2^9\mathbb{Z}$, it is slightly smaller than BJMM. Table 6 displays the workfactors of statistical decoding and BJMM for $n = 25$ and rings larger than 2^9. Especially, we consider the rings $\mathbb{Z}/2^{10}\mathbb{Z}$, $\mathbb{Z}/2^{11}\mathbb{Z}$ and $\mathbb{Z}/2^{12}\mathbb{Z}$. As we can see, a ring of size 2^{10} is sufficiently large for statistical decoding without the precomputation to be more efficient than BJMM. Moreover, in $\mathbb{Z}/2^{12}\mathbb{Z}$, even the full statistical decoding is faster than BJMM. In fact, the statistical decoding algorithm seems to improve when the size of the ring increases.

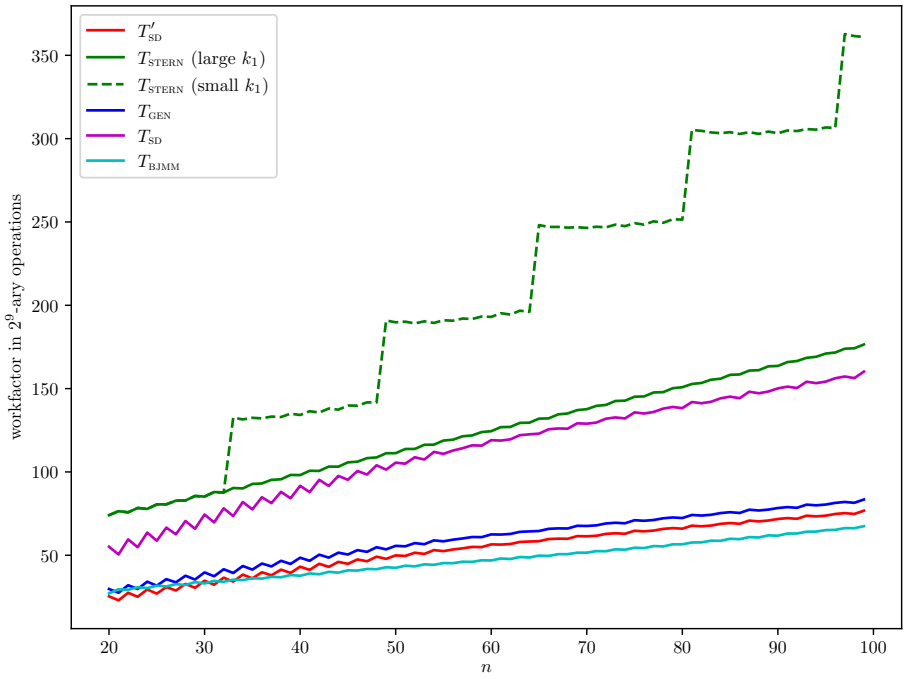


Fig. 12: Workfactors for statistical decoding and Stern algorithms over $(\mathbb{Z}/2^9\mathbb{Z})^n$ embedded with the Lee metric.

Algorithm \ Ring	$\mathbb{Z}/2^{10}\mathbb{Z}$	$\mathbb{Z}/2^{11}\mathbb{Z}$	$\mathbb{Z}/2^{12}\mathbb{Z}$
BJMM	37.206	43.981	49.833
SD	45.169	36.636	31.547
SD'	20.200	15.974	13.637

Table 6: Workfactors of statistical decoding and BJMM for $n = 25$ over the rings $\mathbb{Z}/2^{10}\mathbb{Z}$, $\mathbb{Z}/2^{11}\mathbb{Z}$ and $\mathbb{Z}/2^{12}\mathbb{Z}$. The algorithm SD' corresponds to statistical decoding without accounting for the precomputation time.

6 Conclusion

We have shown that statistical decoding can be extended to finite rings of the form $\mathbb{Z}/m\mathbb{Z}$ over the Lee metric. Moreover, changing the metric does not worsen the efficiency of the algorithm, on the contrary, it reduces to an instance of statistical decoding over the Hamming metric with an error of smaller weight. Exchanging a field for a ring, even a larger one, does not impact much the algo-

rithm. This has to be expected since statistical decoding depends on the number of erroneous positions and not on the entries themselves and the algorithm does not profit from the field structure. On the other hand, ISD algorithms are more affected by an increase of the structure size, and thus, by fixing n, we can find m_0 such that for all $m \geq m_0$, statistical decoding is superior to BJMM.

Note that the main goal of this thesis was to analyze how statistical decoding performs over $\mathbb{Z}/m\mathbb{Z}$ embedded with the Lee metric and how it may impact code-based cryptography. Small improvement and implementation details such as parallelization are not discussed in this thesis. Also, the computation of $\mathcal{H}_w$ has not been discussed in detail and we only gave a naive algorithm to compute it.

This thesis presents natural transformations of statistical decoding but in any way, these algorithms benefit from the Lee metric structure. We think that it would be possible to improve the naive extension of statistical decoding over the Lee metric. For example, one might be interested in the idea of *restricted balls* [3] that benefits from the Lee metric structure. It would be also interesting to analyze how we would apply Overbeck's improvement to the Lee metric [34]. The latter generalizes the set $\mathcal{H}_w$ to multiple values of w varying in a small range of integers.

Another important improvement would be the adapt statistical decoding 2.0 [9] to the Lee metric. This new algorithm drastically ameliorates the performance of classical statistical decoding. Since statistical decoding 2.0 is a natural generalization of statistical decoding, we think that the improvement that we get by changing the metric to the Lee one will still be present over statistical decoding 2.0. For the same reason, we believe that extending the algorithm over finite rings does not impact its running time.

A plaintext recovery attack over the Lee-McEliece was proposed in [22]. Their attack is based on a reduction from a cryptosystem over $\mathbb{Z}/p^s\mathbb{Z}$ to another one over $\mathbb{Z}/p^j\mathbb{Z}$ for any choice of $1 \leq j < s$. We may combine this attack with statistical decoding over the Lee metric. It would be interesting to spend more time on the design and optimization of an efficient algorithm that combines the multiple improvements discussed so far in order to deduce a concrete public key size and compare it with the Hamming metric. We briefly list all possible improvements we have discussed so far:

1. Overbeck's improvement [34];
2. Statistical decoding 2.0 [9];
3. Restricted ball [3];
4. Improve over the naive linear algebra algorithms of the statistical decoding algorithm;
5. Combine the plaintext attack from [22];
6. Study better algorithm for the computation of $\mathcal{H}_w$.

References

1. Nist submissions - pqc wiki. `https://pqc-wiki.fau.edu/w/Special:DatabaseHome`. Accessed: 2023-06-08.

2. Morton Abramson. Combinations, compositions and occupancy problems.
3. Jessica Bariffi, Karan Khathuria, and Violetta Weger. Information set decoding for lee-metric codes using restricted balls. In Jean-Christophe Deneuville, editor, *Code-Based Cryptography*, pages 110–136, Cham, 2023. Springer Nature Switzerland.
4. Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2n/20: How $1 + 1 = 0$ improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 520–536, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
5. E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.
6. E.R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill series in systems science. World Scientific, 2015.
7. Daniel J. Bernstein. *Introduction to post-quantum cryptography*, pages 1–14. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
8. Eimear Byrne and Violetta Weger. Bounds in the lee metric and optimal codes. *Finite Fields and Their Applications*, 87:102151, 2023.
9. Kevin Carrier, Thomas Debris-Alazard, Charles Meyer-Hilfiger, and Jean-Pierre Tillich. Statistical decoding 2.0: Reducing decoding to lpn, 2022.
10. Thomas Debris-Alazard and Jean-Pierre Tillich. Statistical decoding, 2017.
11. Hang Dinh, Cristopher Moore, and Alexander Russell. Mceliece and niederreiter cryptosystems that resist quantum fourier sampling attacks. In Phillip Rogaway, editor, *Advances in Cryptology – CRYPTO 2011*, pages 761–779, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
12. Jean-Charles Faugère, Valérie Gauthier-Umanã, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high rate mceliece cryptosystems. In *2011 IEEE Information Theory Workshop*, pages 282–286, 2011.
13. V. D. Goppa. Codes associated with divisors. *Problemy Peredachi Informatsii*, 13:33–39, 1977.
14. Marcus Greferath. *An Introduction to Ring-Linear Coding Theory*, pages 219–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
15. Yann Hamdaoui and Nicolas Sendrier. A non asymptotic analysis of information set decoding. *IACR Cryptol. ePrint Arch.*, 2013:162, 2013.
16. R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950.
17. Juncheol Han. The general linear group over a ring. *Bulletin of the Korean Mathematical Society*, 43(3):619–626, 08 2006.
18. Thomas Honold, Zentrum Mathematik, and Ivan Landjev. Linear codes over finite chain rings. *The Electronic Journal of Combinatorics*, 7:R11–R11, 4 2000.
19. Anna-Lena Horlemann-Trautmann and Violetta Weger. Information set decoding in the lee metric with applications to cryptography, 2020.
20. A. Al Jabri. A statistical decoding algorithm for general linear block codes. In Bahram Honary, editor, *Cryptography and Coding*, pages 1–8, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.
21. Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography, Second Edition*. Chapman & Hall/CRC, 2nd edition, 2014.
22. Terry Lau and Chik How Tan. On the design and security of lee metric mceliece cryptosystems. *Designs, Codes and Cryptography*, 90, 03 2022.

23. P. J. Lee and E. F. Brickell. An observation on the security of mceliece's public-key cryptosystem. In D. Barstow, W. Brauer, P. Brinch Hansen, D. Gries, D. Luckham, C. Moler, A. Pnueli, G. Seegmüller, J. Stoer, N. Wirth, and Christoph G. Günther, editors, *Advances in Cryptology — EUROCRYPT '88*, pages 275–280, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg.
24. Arjen K. Lenstra and Eric R. Verheul. Selecting cryptographic key sizes. In Hideki Imai and Yuliang Zheng, editors, *Public Key Cryptography*, pages 446–465, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
25. J.S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, 1988.
26. Yuan Xing Li, R.H. Deng, and Xin Mei Wang. On the equivalence of mceliece's and niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–273, 1994.
27. Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{O}(20.054n)$. In *Proceedings of the 17th International Conference on The Theory and Application of Cryptology and Information Security*, ASIACRYPT'11, page 107–124, Berlin, Heidelberg, 2011. Springer-Verlag.
28. R. J. McEliece. A Public-Key Cryptosystem Based On Algebraic Coding Theory. *Deep Space Network Progress Report*, 44:114–116, January 1978.
29. Michele Mosca. Cybersecurity in an era with quantum computers: Will we be ready? *IEEE Security & Privacy*, 16(5):38–41, 2018.
30. M.T.L.B. An introduction to probability theory and its applications. by william feller [2nd edn. pp. xv 461. new york: John wiley and sons; london: Chapman and hall. 1957. 86s.]. *Journal of the Institute of Actuaries*, 84(2):232–234, 1958.
31. Robert Niebuhr. Statistical decoding of codes over $\mathbb{F}_q$. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, pages 217–227, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
32. Robert Niebuhr, Mohammed Meziani, Stanislav Bulygin, and Johannes Buchmann. Selecting parameters for secure mceliece-based cryptosystems. *IACR Cryptology ePrint Archive*, 2010:271, 01 2010.
33. H. NIEDERREITER. Knapsack-type cryptosystems and algebraic coding theory. *Prob. Contr. Inform. Theory*, 15(2):157–166, 1986.
34. R. Overbeck. Statistical decoding revisited. In Lynn Margaret Batten and Reihaneh Safavi-Naini, editors, *Information Security and Privacy*, pages 283–294, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
35. Edoardo Persichetti. On the cca2 security of mceliece in the standard model. In Joonsang Baek, Willy Susilo, and Jongkil Kim, editors, *Provable Security*, pages 165–181, Cham, 2018. Springer International Publishing.
36. Christiane Peters. Information-set decoding for linear codes over fq. In Nicolas Sendrier, editor, *Post-Quantum Cryptography*, pages 81–94, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
37. Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Inf. Theory*, 8:5–9, 1962.
38. Steven Roman. Advanced linear algebra. 135, 1992.
39. Ron Roth. *Introduction to Coding Theory*. Cambridge University Press, USA, 2006.
40. Keisuke Shiromoto. Singleton bounds for codes over finite rings. *Journal of Algebraic Combinatorics*, 12:95–99, 01 2000.

41. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, oct 1997.
42. WOLFGANG STADJE. The residues modulo m of products of random integers. *Combinatorics, Probability and Computing*, 11(5):529–540, 2002.
43. Jacques Stern. A method for finding codewords of small weight. In Gérard Cohen and Jacques Wolfmann, editors, *Coding Theory and Applications*, pages 106–113, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg.
44. David Wagner. A generalized birthday problem. In Moti Yung, editor, *Advances in Cryptology — CRYPTO 2002*, pages 288–304, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.
45. Violetta Weger. *Information Set Decoding in the Lee Metric and the Local to Global Principle for Densities*. PhD thesis, 2020.
46. Violetta Weger, Massimo Battaglioni, Paolo Santini, Franco Chiaraluce, Marco Baldi, and Edoardo Persichetti. Information set decoding of lee-metric codes over finite rings, 2021.
47. Violetta Weger, Karan Khathuria, Anna-Lena Horlemann, Massimo Battaglioni, Paolo Santini, and Edoardo Persichetti. On the hardness of the lee syndrome decoding problem, 2022.