Advances in Mathematics of Communications
Volume 15, No. 4, 2021, 677–699
doi:10.3934/amc.2020089

INFORMATION SET DECODING IN THE LEE METRIC WITH APPLICATIONS TO CRYPTOGRAPHY

Anna-Lena Horlemann-Trautmann
Faculty of Mathematics and Statistics
University of St. Gallen
Bodanstr. 6, St. Gallen, Switzerland

Violetta Weger*
Institute of Mathematics
University of Zurich
Winterthurerstrasse 190, 8057 Zurich, Switzerland

(Communicated by Ferruh Özbudak)

Abstract. We convert Stern's information set decoding (ISD) algorithm to the ring $\mathbb{Z}/4\mathbb{Z}$ equipped with the Lee metric. Moreover, we set up the general framework for a McEliece and a Niederreiter cryptosystem over this ring. The complexity of the ISD algorithm determines the minimum key size in these cryptosystems for a given security level. We show that using Lee metric codes can substantially decrease the key size, compared to Hamming metric codes. In the end we explain how our results can be generalized to other Galois rings $\mathbb{Z}/p^s\mathbb{Z}$.

1. Introduction

The hardness of decoding random linear codes is at the heart of any code-based public key cryptosystem. Information set decoding (ISD) algorithms are the main method for decoding random linear codes in the Hamming metric, whenever the problem has only a few solutions. An ISD algorithm is given a corrupted codeword and recovers the message or equivalently finds the error vector. Such algorithms are often formulated via the parity check matrix, since it is enough to find a vector of a certain weight which has the same syndrome as the corrupted codeword – this problem is also referred to as the syndrome decoding problem. ISD algorithms over the binary are based on a decoding algorithm proposed by Prange [37] in 1962 and the main structure of the variants do not change much from the original: as a first step one chooses an information set, then Gaussian elimination brings the parity check matrix into a standard form and, assuming that the errors are outside of the information set, these row operations on the syndrome will recover the error vector, if the weight does not exceed the given error correction capacity.

ISD algorithms are of immense importance when proposing a code-based cryptosystem. The idea of using linear codes in public key cryptography was first formulated by Robert McEliece [31], in 1978. In the McEliece cryptosystem the private key is the generator matrix of a linear code with an efficient decoding algorithm.

2020 *Mathematics Subject Classification:* Primary: 11T71; Secondary: 68P30.
Key words and phrases: Coding theory, Cryptography, McEliece system, Ring-linear codes, Lee metric.

* Corresponding author: Violetta Weger.

The public key is a scrambled and disguised version of the generator matrix, such that the private key (and hence the decoding algorithm) is not reconstructable from the public key. The message is encrypted by encoding it with the generator matrix and adding a random error of prescribed Hamming weight. The owner of the private key can recover the message by inverting the disguising function and using the efficient decoding algorithm. On the other hand, if the secret code is hidden well enough, an adversary who wants to break the system encounters the decoding problem of a random linear code, since the public code looks random to him. The best the adversary can do is hence to use the best generic decoding algorithm for random linear codes, which currently are ISD algorithms. ISD algorithms hence do not break a code-based cryptosystem but they determine the choice of secure parameters.

One of the main drawbacks of classical code-based cryptosystems are the public key sizes. To reduce these key sizes, over the last years, many variants of code-based cryptosystems have been proposed that use codes in the rank-metric, instead of the Hamming metric. This raises the question if other metrics can be useful, as well. This is why we study codes in the Lee metric for code-based cryptography in this paper, and show that, for theoretical code parameters[1], the Lee metric can also lead to a substantial reduction of the public key size.

We will focus on codes that are defined over integer residue rings $\mathbb{Z}_m := \mathbb{Z}/m\mathbb{Z}$, for some integer $m > 1$, equipped with the Lee metric. In particular, we are going to use ring-linear codes, which are defined to be $\mathbb{Z}_m$-submodules of $\mathbb{Z}_m^n$. We especially focus on quaternary codes, which are defined over $\mathbb{Z}_4$, since this case has been studied the most in the coding theory literature. In general, ring-linear codes were first mentioned by Assmus and Mattson in [1], for important results see [8, 9, 21, 33, 39, 40, 41], for a more general overview see [20]. The idea of using ring-linear codes for cryptography (in a quite different setting) first came up in [44].

We note that, although $\mathbb{Z}_4$-linear Lee codes can be represented over $\mathbb{F}_2$, there exists no representation that preserves both the weight and the linearity of the $\mathbb{Z}_4$-code over $\mathbb{F}_2$. Thus the known results over $\mathbb{F}_2$ cannot be used for the Lee metric. This is why this paper presents the adaption of ISD algorithms over the binary [42] to $\mathbb{Z}_4$ and a general form of the McEliece and Niederreiter cryptosystems over $\mathbb{Z}_4$. The complexity of the ISD algorithm then determines a minimum public key size for a given security level of these cryptosystems. The paper is structured as follows: in Section 2 we introduce the theory of ring-linear codes, especially Lee codes, two ISD algorithms over the binary [42] and the notations and concepts involved in the algorithms. In Section 3 we present the adaption of the ISD algorithms over the binary to $\mathbb{Z}_4$, including a complexity analysis. In Section 4 we cover the applications of the ISD algorithm over $\mathbb{Z}_4$ to code-based cryptography by stating the general McEliece and Niederreiter cryptosystems using quaternary codes. In this context we will also investigate the key size of such a cryptosystem using theoretical values for the secret quaternary code regarding 128 bit security against our ISD algorithms over $\mathbb{Z}_4$, from Section 3. In Section 5 we explain briefly how one can generalize the ISD algorithm as well as the McEliece system to other Galois rings $\mathbb{Z}_{p^s}$. We will then conclude this paper in Section 6 and add some open questions and problems.

[1]By theoretical parameters we mean codes attaining the Gilbert-Varshamov bound.

2. Preliminaries

In this section we present the main theory and tools of ring-linear codes, especially Lee codes, as well as some known binary ISD algorithms and the concepts and notations involved.

2.1. Ring-linear coding theory. In traditional finite field coding theory an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ is a linear subspace of $\mathbb{F}_q^n$ of dimension k. One can generalize this by taking a finite ring R instead of $\mathbb{F}_q$.

Let us assume for simplicity that R is commutative, but observe that the following stays true in the noncommutative case.

Definition 2.1. Let k and n be positive integers and let R be a finite ring. $\mathcal{C}$ is called an *R-linear code of length n of type h*, if $\mathcal{C}$ is a submodule of R^n, with $|\mathcal{C}| = h$.

We will restrict to the most preferred case of $\mathbb{Z}_m := \mathbb{Z}/m\mathbb{Z}$, for some $m \in \mathbb{N}$. In particular, we will formulate most of our results for $m = 4$, since this case has been studied the most.

Definition 2.2. We say that $\mathcal{C}$ is a *quaternary linear code* of length n, if $\mathcal{C}$ is an additive subgroup of $\mathbb{Z}_4^n$.

In traditional finite field coding theory we endow $\mathbb{F}_q^n$ with the Hamming metric to define the weight of a codeword wt_H and the distance of codewords d_H. In ring-linear coding theory over $\mathbb{Z}_m$ we could use the Hamming metric, the Lee metric, the homogeneous metric, the Euclidean metric and so on, for an overview see [19]. If we use the Lee metric the corresponding codes are referred to as *Lee codes*.

Definition 2.3. For $x \in \mathbb{Z}_m$ we define the *Lee weight* to be

$$\mathrm{wt}_L(x) = \min\{x, m - x\},$$

similarly for $x \in \mathbb{Z}_m^n$ we define the Lee weight to be the sum of the Lee weights of its coordinates:

$$\mathrm{wt}_L(x) = \sum_{i=1}^{n} \mathrm{wt}_L(x_i).$$

For $x, y \in \mathbb{Z}_m^n$, the *Lee distance* is defined to be

$$d_L(x, y) = \mathrm{wt}_L(x - y).$$

There is a connection between traditional finite field coding theory and $\mathbb{Z}_4$-linear coding theory via the Gray map:

Definition 2.4. The Gray map is an isometry between $(\mathbb{Z}_4, \mathrm{wt}_L)$ and $(\mathbb{F}_2^2, \mathrm{wt}_H)$ and is defined as follows:

$$\begin{aligned} \phi : (\mathbb{Z}_4, \mathrm{wt}_L) &\to (\mathbb{F}_2^2, \mathrm{wt}_H) \\ 0 &\mapsto (0, 0), \\ 1 &\mapsto (0, 1), \\ 2 &\mapsto (1, 1), \\ 3 &\mapsto (1, 0). \end{aligned}$$

The Gray map can be extended componentwise to

$$\overline{\phi} : (\mathbb{Z}_4^n, \mathrm{wt}_L) \to (\mathbb{F}_2^{2n}, \mathrm{wt}_H).$$

Note however, that the Gray map does not preserve linearity, i.e., the image of a quaternary linear code is generally not linear over $\mathbb{F}_2$.

We introduce the following notation: For a vector v of length n, a matrix A with n columns, a code $\mathcal{C}$ of length n and a set $I \subset \{1,\dots,n\}$ we denote by v_I the projection of v to its coordinates indexed by I, and by A_I the columns of A indexed by I. Analogously we define $\mathcal{C}_I := \{v_I \mid v \in \mathcal{C}\}$.

We will use the following definition of information set, since it fits perfectly in the context of ring-linear codes:

Definition 2.5. For a code $\mathcal{C}$ over $\mathbb{F}_q$ of length n and dimension k, we call a set $I \subseteq \{1,\dots,n\}$ of size k an *information set* if $\mid \mathcal{C}_I \mid = \mid \mathcal{C} \mid$.

Similarly, we define quaternary information sets for quaternary codes as follows:

Definition 2.6. For a code $\mathcal{C}$ over $\mathbb{Z}_4$ of length n and type $4^{k_1}2^{k_2}$, we call a set $I \subseteq \{1,\dots,n\}$ of size k_1+k_2 a *(quaternary) information set* if $\mid \mathcal{C}_I \mid = \mid \mathcal{C} \mid$.

The following proposition defines the quaternary systematic form of the generator matrix and the parity check matrix of a quaternary code.

Proposition 1 ([21])**.** *Let $\mathcal{C}$ be a quaternary linear code of length n and type $4^{k_1}2^{k_2}$. Then $\mathcal{C}$ is permutation equivalent to a code having the $(k_1+k_2)\times n$ generator matrix*

$$G = \begin{pmatrix} \mathrm{Id}_{k_1} & A & B \\ 0 & 2\,\mathrm{Id}_{k_2} & 2C \end{pmatrix}, \tag{1}$$

where $A \in \mathbb{Z}_2^{k_1\times k_2}$, $B \in \mathbb{Z}_4^{k_1\times(n-k_1-k_2)}$, $C \in \mathbb{Z}_2^{k_2\times(n-k_1-k_2)}$, for some $k_1,k_2 \in \mathbb{N}_0$.

A parity check matrix of $\mathcal{C}$ is the corresponding permutation of the $(n-k_1)\times n$ matrix

$$H = \begin{pmatrix} -B^\top - C^\top A^\top & C^\top & \mathrm{Id}_{n-k_1-k_2} \\ 2A^\top & 2\,\mathrm{Id}_{k_2} & 0 \end{pmatrix} =: \begin{pmatrix} D & E & \mathrm{Id}_{n-k_1-k_2} \\ 2F & 2\,\mathrm{Id}_{k_2} & 0 \end{pmatrix}, \tag{2}$$

where $D \in \mathbb{Z}_4^{(n-k_1-k_2)\times k_1}$, $E \in \mathbb{Z}_2^{(n-k_1-k_2)\times k_2}$, $F \in \mathbb{Z}_2^{k_2\times k_1}$.

If we have a generator matrix of the form (1), to get a unique encoding, the messages need to be of the form $m = (m_1, m_2)$, where $m_1 \in \mathbb{Z}_4^{k_1}$ and $m_2 \in \mathbb{Z}_2^{k_2}$. Encoding is done as follows:

$$(m_1,m_2)\begin{pmatrix} \mathrm{Id}_{k_1} & A & B \\ 0 & 2\,\mathrm{Id}_{k_2} & 2C \end{pmatrix} = \begin{pmatrix} m_1^\top \\ (m_1A+2m_2)^\top \\ (m_1B+2m_2C)^\top \end{pmatrix} = \begin{pmatrix} c_1^\top \\ c_2^\top \\ c_3^\top \end{pmatrix}.$$

Hence the codewords are of the form $c = (c_1,c_2,c_3)$, where $c_1 \in \mathbb{Z}_4^{k_1}, c_2 \in \mathbb{Z}_4^{k_2}$ and $c_3 \in \mathbb{Z}_4^{n-k_1-k_2}$.

For the syndrome of a codeword $c = (c_1,c_2,c_3)$ we get

$$\begin{pmatrix} D & E & \mathrm{Id}_{n-k_1-k_2} \\ 2F & 2\,\mathrm{Id}_{k_2} & 0 \end{pmatrix}\begin{pmatrix} c_1^\top \\ c_2^\top \\ c_3^\top \end{pmatrix} = \begin{pmatrix} Dc_1^\top + Ec_2^\top + c_3^\top \\ 2Fc_1^\top + 2c_2^\top \end{pmatrix} = \begin{pmatrix} s_1^\top \\ 2s_2^\top \end{pmatrix}.$$

The syndromes $s = (s_1, 2s_2)$ are such that $s_1 \in \mathbb{Z}_4^{n-k_1-k_2}$ and $s_2 \in \mathbb{Z}_2^{k_2}$.

To compute the number of vectors in $\mathbb{Z}_4^n$ having Lee weight w, we have to sum over all choices of i entries having Lee weight 2, of course only until $\lfloor w/2 \rfloor$. For the rest of the $n-i$ entries we are missing a Lee weight of $w-2i$. We will achieve this with entries of Lee weight 1, where for each of the $w-2i$ entries, there are two

choices: either 1 or 3. We will introduce the following notation for the amount of these vectors:

$$c(n,w) := \sum_{i=0}^{\lfloor w/2 \rfloor} \binom{n}{i}\binom{n-i}{w-2i}2^{w-2i}.$$

With the Gray isometry we have that the number of vectors in $\mathbb{Z}_4^n$ having Lee weight w is the same as the number of vectors in $\mathbb{F}_2^{2n}$ having Hamming weight w, which is simply given by $\binom{2n}{w}$. Note that one can also check that

$$c(n,w) = \sum_{i=0}^{\lfloor w/2 \rfloor} \binom{n}{i}\binom{n-i}{w-2i}2^{w-2i} = \binom{2n}{w}. \tag{3}$$

With this we can easily derive an analogue of the binary Gilbert-Varshamov bound for quaternary codes.

Proposition 2 (Theorem 13.73, [3]). *Let n and d be positive integers. There exists a linear binary code $\mathcal{C}$ of length n and minimum Hamming distance d, such that*

$$|\,\mathcal{C}\,| \geq \frac{2^n}{\sum_{j=0}^{d-1}\binom{n}{j}}.$$

Furthermore there exists a linear quaternary code $\mathcal{C}$ of length n and minimum Lee distance d, such that

$$|\,\mathcal{C}\,| \geq \frac{4^n}{(\sum_{j=0}^{d-1}\binom{2n}{j}-1)3+1}.$$

2.2. Information set decoding algorithms. Many ISD algorithms and improvements have been suggested to Prange's simplest form of ISD (see for example [11, 13, 14, 16, 26, 45]); in historical order the proposed ISD algorithms are by Prange [37], Leon [28], Lee-Brickell [27], Stern [42], Canteaut and Chabaud [12], Finiasz and Sendrier [18], Bernstein, Lange and Peters [6], May, Meurer and Thomae [29], Becker, Joux, May and Meurer [2] and the latest improvement is by May and Ozerov [30].

All of the above mentioned ISD algorithms were proposed over the binary field. However, with new variants of the McEliece cryptosystem proposed over general finite fields, some of the mentioned ISD algorithms have been generalized to $\mathbb{F}_q$: Coffey and Goodman [15] generalized Prange's algorithm to $\mathbb{F}_q$, in [35] Peters generalized the algorithms by Lee-Brickell and Stern. Niebuhr, Persichetti, Cayrel, Bulygin and Buchmann [34] generalized the algorithm of Finiasz-Sendrier with efficiency improvements by using partial knowledge of attackers to a general finite field. In [24] Interlando, Khathuria, Rohrer, Rosenthal and Weger generalized the ball-collision algorithm by Bernstein, Lange and Peters to $\mathbb{F}_q$. In [23] Hirose generalized the May-Ozerov algorithm to $\mathbb{F}_q$. And Meurer generalized the algorithm of Becker, Joux, May and Meurer in [32].

The general idea of ISD algorithms is to guess an information set $I \subset \{1,\dots n\}$ of size k and the right distribution of the error vector corresponding to this information set, such that we can recover the message from this information set. In the algorithms we consider the information set I will be chosen randomly in each outer loop of the algorithm. Nevertheless we want to note that there is a slightly smarter way to do so, see [12], by reusing some elements of I in the next iteration and only adding missing elements. For simplicity, we will just use a random choice.

Once we have chosen an information set I, we need to guess the error vector having the assumed weight distribution. In the binary case this means we just have to guess the locations of the errors. In Lee-Brickell's algorithm [27], the distribution of the error vector is assumed to be w in the information set and $t-w$ outside the information set. In Stern's algorithm the error vector has weight $2v$ in the information set coordinates; moreover, these $2v$ errors are assumed to be located in two disjoint coordinate sets, both having weight v. In addition, we assume that there is a zero-window of size ℓ, where no errors are allowed. The remaining error weight of $t-2v$ is found in the remaining $n-k-\ell$ coordinates.

The average complexity of ISD algorithms is given by the cost of one iteration times the average number of iterations needed, which is given by the inverted success probability. Note that the success criterion is to choose the correct weight distribution of the error vector. The success probability of having correctly chosen w errors in k coordinates over all vectors having length n and Hamming weight t is given by

$$\binom{k}{w}\binom{n}{t}^{-1}.$$

While on classical computers ISD attacks with a high cost of one iteration but a low number of iterations outperform ISD attacks with a low cost of one iteration and a high number of iterations, this is not the case for quantum computers. In fact: in [4] it was observed that using Grover's algorithm within ISD attacks reduces the number of iterations needed, thus when using a quantum computer ISD attacks with a low cost of one iteration, such as Lee-Brickell's algorithm, might outperform ISD attacks a low number of iterations, such as Stern's algorithm. This is why we will adapt both, Lee-Brickell's algorithm and Stern's algorithm, to the Lee metric.

We will start with explaining the ISD algorithm of Lee-Brickell [27] over the binary field with respect to the Hamming distance. For this we are going to use the following notation. For $S \subset \{1,\dots,n\}$ of size ℓ, we denote by $\mathbb{F}_2^n(S)$ the vectors living in $\mathbb{F}_2^n$ having support in S. The projection of $x \in \mathbb{F}_2^n(S)$ to $\mathbb{F}_2^\ell$ is denoted by $\pi_S(x)$. On the other hand we denote by $\sigma_S(x)$ the canonical embedding of $x \in \mathbb{F}_2^\ell$ to $\mathbb{F}_2^n(S)$. We are given the parity check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, the amount t of errors we can correct and the syndrome $s \in \mathbb{F}_2^{n-k}$. We want to find a vector $e \in \mathbb{F}_2^n$, such that $\mathrm{wt}_H(e) = t$ and $He^\top = s$. The algorithm is formulated in Algorithm 1.

Algorithm 1 Lee-Brickell's algorithm over $\mathbb{F}_2$

Input: The parity check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, $s \in \mathbb{F}_2^{n-k}$ and the positive integer $w \in \mathbb{Z}$, such that $w \le t$, $w \le k$ and $t - w \le n - k$.
Output: $e \in \mathbb{F}_2^n$ with $He^\top = s^\top$ and $wt_H(e) = t$.

1: Choose an information set $I \subset \{1,...,n\}$ of size k, let $J = \{1,...,n\} \setminus I$.
2: Find an invertible matrix $U \in \mathbb{F}_2^{(n-k)\times(n-k)}$ such that $(UH)_J = \mathrm{Id}_{n-k}$ and $(UH)_I = A$, where $A \in \mathbb{F}_2^{(n-k)\times k}$.
3: Compute $Us^\top = s'^\top$.
4: **for** each $e_1 \in \mathbb{F}_2^n(I)$, with $wt_H(e_1) = w$ **do**
5: **if** $wt_H(s' + \pi_I(e_1)A^\top) = t - w$: **then**
6: Output: $e = e_1 + \sigma_J(s' + \pi_I(e_1)A^\top)$.
7: Start over with Step 1 and a new selection of I.

To illustrate the algorithm, assume that the information set is $I = \{1, \ldots, k\}$. We get

$$UHe^\top = \begin{pmatrix} A & \mathrm{Id}_{n-k} \end{pmatrix} \begin{pmatrix} e_1^\top \\ e_2^\top \end{pmatrix} = s'^\top = Us^\top,$$

where $A \in \mathbb{F}_2^{(n-k)\times k}$ and $e_1 \in \mathbb{F}_2^k, e_2 \in \mathbb{F}_2^{n-k}$. From this we get the condition

$$e_1 A^\top + e_2 = s'.$$

The part e_1 is chosen to have weight w, and the part e_2 is chosen such that its support is disjoint from e_1, it has the remaining weight $t - w$, and $e_2 = s' + e_1 A^\top$.

Remark 1. In all the following complexity analyses we will use schoolbook long multiplication with a computational complexity of n^2 operations for inputs of length n. We remark that faster multiplication algorithms are known and can be used in our ISD algorithms; however, we refrain from using them since they would not make a substantial difference in our analyses, but on the other hand make the formulas more complicated.

Theorem 2.7. *The average number of bit operations Algorithm 1 needs is approximately*

$$\left(\binom{k}{w}\binom{n-k}{t-w}\right)^{-1}\binom{n}{t} \cdot \left[(n-k)^2(n+1) + \binom{k}{w}(w+1)(n-k)\right].$$

Proof. As a first step we need to find the systematic form of the permuted parity check matrix, and the corresponding syndrome form. As a broad estimate we use the complexity of computing $U[H \mid s^\top]$, which takes approximately $(n-k)^2(n+1)$ bit operations.

For all $e_1 \in \mathbb{F}_2^n(I)$ having $wt_H(e_1) = w$, which are $\binom{k}{w}$ many, we have to check, if the weight of $e_2 = s' + \pi_I(e_1)A^\top$ is $t - w$, hence this step costs $(w+1)(n-k)$ bit operations.

The success probability is given by having chosen the correct weight distribution of the error vector, i.e., in the information set the weight w and in J the missing weight $t - w$:

$$\binom{k}{w}\binom{n-k}{t-w}\binom{n}{t}^{-1}.$$

Hence, the overall cost of this algorithm is as claimed. □

In the following we explain Stern's algorithm over the binary field with respect to the Hamming distance. We will use a formulation of the algorithm, which matches the ball-collision formulation in [6]. The two algorithms differ in the zero-window of size ℓ: in Stern's algorithm no error is allowed in the zero-window, whereas in the ball-collision algorithm, this window is split into Y_1, Y_2 and q_1, q_2 errors are allowed respectively. Even though the asymptotic complexity of the ball-collision algorithm is smaller, for concrete parameters it turns out that, in most of the cases, $q_1 = q_2 = 0$ is the most efficient choice, therefore we will generalize Stern's algorithm to $\mathbb{Z}_4$ in this paper. Nevertheless, using the (ball-)collision formulation allows us to use improvements and speed ups, some of which will be explained in the following, together with their complexities:

1. The concept of *intermediate sums* presented in [6] is important whenever one wants to do a computation for all vectors in a certain space. Consider, for example, the setting where we are given a binary $k \times n$ matrix A and want to

compute $Ax^\top$ for all $x \in \mathbb{F}_2^n$, of weight w. This would usually cost $k(w-1)$ additions and w multiplications, for each $x \in \mathbb{F}_2^n$. But if we first compute $Ax^\top$, where x has weight one, this only outputs the corresponding column of A and has no costs. From there we can compute the sums of two columns of A, there are $\binom{n}{2}$ many of these sums and each one costs k additions. From there we can compute all sums of three columns of A, which are $\binom{n}{3}$ many. Using the sums of two columns, we only need to add one more column costing k additions. Proceeding in this way, until one reaches the weight w, computing $Ax^\top$ for all $x \in \mathbb{F}_2^n$ of weight w, costs $k(L(n,w)-n)$ bit operations, where

$$L(n,w) := \sum_{i=1}^{w} \binom{n}{i}.$$

Note that we need to take away the cost of the weight one vectors, since they are for free.

2. The next concept, called *early abort* (also presented in [6]), is important whenever a computation is done while checking the weight of the result. For example one wants to compute $x+y$, where $x, y \in \mathbb{F}_2^n$, which usually costs n additions, but we only proceed in the algorithm if $wt_H(x+y) = t$. Hence we compute and check the weight simultaneously, and if the weight of the partial solution exceeds t, we do not need to continue. For the Hamming weight one expects a randomly chosen bit to have weight 1 with probability $\frac{1}{2}$, hence after $2t$ we should reach the wanted weight t, and after $2(t+1)$ we should exceed the weight t. Hence on average we expect to compute only $2(t+1)$ many bits of the solution, before we can abort.
3. The third concept is the idea of using *collisions*. Instead of going through all possible error vectors of weight $2v$, fulfilling certain properties, we can split this process. For this we consider vectors of weight v in one set S, and vectors with disjoint weight v in another set T, such that two vectors in the intersection of S and T determine the final error vector. The exact definition of the two sets and the properties the vectors need to fulfill will become clearer when we describe the actual algorithm, but we can assume that we need to find x, y from some $S \subseteq \mathbb{F}_2^n$ and $T \subseteq \mathbb{F}_2^n$, respectively, such that $Ax^\top = By^\top + s^\top$ for some prescribed $A, B \in \mathbb{F}_2^{k \times n}$ and $s \in \mathbb{F}_2^k$. Assuming that x, y are uniformly distributed, the average amount of collisions is given by

$$| S | \cdot | T | \cdot 2^{-n}.$$

The assumption of a uniform distribution is commonly used, and justified e.g. in [6] and references therein.

The setting for the algorithm is as follows: We are given the parity check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, the amount t of errors we can correct and the syndrome $s \in \mathbb{F}_2^{n-k}$. We want to find a vector $e \in \mathbb{F}_2^n$, such that $\mathrm{wt}_H(e) = t$ and $He^\top = s^\top$. We are going to use all the ideas mentioned above. Stern's algorithm over the binary in the collision formulation is given in Algorithm 2. Note that, without formulating it in detail, the concept of intermediate sums is used in lines 6 and 7, and the concept of early abort is used in line 10. The collision is used in lines 8 and 9, since it is the same a in both cases.

Algorithm 2 Collision ISD (Stern's algorithm) over $\mathbb{F}_2$

Input: The parity check matrix $H \in \mathbb{F}_2^{(n-k)\times n}$, $s \in \mathbb{F}_2^{n-k}$ and the non-negative integers $v, m_1, m_2, \ell \in \mathbb{Z}$, such that $k = m_1 + m_2$, $v \leq m_1, m_2$, $\ell \leq n-k$ and $t - 2v \leq n - k - \ell$.
Output: $e \in \mathbb{F}_2^n$ with $He^\top = s^\top$ and $wt_H(e) = t$.

1: Choose an information set $I \subset \{1, ..., n\}$ of size k.
2: Choose a set $J \subset \{1, \ldots, n\} \setminus I$ of size ℓ.
3: Choose a uniform random partition of I into disjoint sets X and Y of size m_1 and $m_2 = k - m_1$, respectively.
4: Find an invertible matrix $U \in \mathbb{F}_2^{(n-k)\times(n-k)}$ such that $(UH)_{I^C} = \mathrm{Id}_{n-k}$ and $(UH)_I = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$, where $A_1 \in \mathbb{F}_2^{\ell\times k}$ and $A_2 \in \mathbb{F}_2^{(n-k-\ell)\times k}$.
5: Compute $Us^\top = \begin{pmatrix} s_1^\top \\ s_2^\top \end{pmatrix}$ with $s_1 \in \mathbb{F}_2^\ell$ and $s_2 \in \mathbb{F}_2^{n-k-\ell}$.
6: Compute the set S consisting of all pairs $(\pi_I(e_X)A_1^\top, e_X)$, where $e_X \in \mathbb{F}_2^n(X)$, $wt_H(e_X) = v$.
7: Compute the set T consisting of all pairs $(\pi_I(e_Y)A_1^\top + s_1, e_Y)$, where $e_Y \in \mathbb{F}_2^n(Y)$, $wt_H(e_Y) = v$.
8: **for** each $(a, e_X) \in S$ **do**
9: **for** each $(a, e_Y) \in T$ **do**
10: **if** $wt_H(\pi_I(e_X + e_Y)A_2^\top + s_2) = t - 2v$: **then**
11: Output: $e = e_X + e_Y + \sigma_J(\pi_I(e_X + e_Y)A_2^\top + s_2)$.
12: Start over with Step 1 and a new selection of I.

Let us illustrate the algorithm in the easiest situation, where the information set is $I = \{1, \ldots, k\}$ and the zero window is $\{k+1, \ldots, k+\ell\}$. We get

$$UHe^\top = \begin{pmatrix} A_1 & \mathrm{Id}_\ell & 0 \\ A_2 & 0 & \mathrm{Id}_{n-k-\ell} \end{pmatrix} \begin{pmatrix} e_1^\top \\ 0 \\ e_2^\top \end{pmatrix} = \begin{pmatrix} s_1^\top \\ s_2^\top \end{pmatrix} = Us^\top,$$

where $A_1 \in \mathbb{F}_2^{\ell\times k}$, $A_2 \in \mathbb{F}_2^{(n-k-\ell)\times k}$ and $e_1 \in \mathbb{F}_2^k, e_2 \in \mathbb{F}_2^{n-k-\ell}, s_1 \in \mathbb{F}_2^\ell, s_2 \in \mathbb{F}_2^{n-k-\ell}$. From this we get the conditions

$$\begin{aligned} e_1 A_1^\top &= s_1, \\ e_1 A_2^\top + e_2 &= s_2. \end{aligned}$$

The part e_1 is chosen to be $\pi_I(e_X + e_Y)$ such that it has weight $2v$, and with the collision we ensure that the first condition is satisfied. The part e_2 is chosen such that its support is disjoint from e_1, it has the remaining weight $t - 2v$, and the second condition is satisfied, i.e., $e_2 = \pi_I(e_X + e_Y)A_2^\top + s_2$. Therefore

$$\begin{aligned} UHe^\top &= \begin{pmatrix} A_1 & \mathrm{Id}_\ell & 0 \\ A_2 & 0 & \mathrm{Id}_{n-k-\ell} \end{pmatrix} \begin{pmatrix} \pi_I(e_X + e_Y)^\top \\ 0 \\ A_2\pi_I(e_X + e_Y)^\top + s_2^\top \end{pmatrix} \\ &= \begin{pmatrix} A_1\pi_I(e_X + e_Y)^\top \\ A_2\pi_I(e_X + e_Y)^\top + A_2\pi_I(e_X + e_Y)^\top + s_2^\top \end{pmatrix} = \begin{pmatrix} s_1^\top \\ s_2^\top \end{pmatrix} = Us^\top, \end{aligned}$$

i.e., $e = (e_1, 0, e_2)$ fulfills $He^\top = s^\top$ and $wt_H(e) = t$.

Theorem 2.8. *The average number of bit operations Algorithm 2 needs, is*

$$\left(\binom{m_1}{v}\binom{m_2}{v}\binom{n-k-\ell}{t-2v}\right)^{-1}\binom{n}{t}\cdot\big[(n-k)^2(n+1)+\ell(L(m_1,v)-m_1)$$
$$+\ell\left(L(m_2,v)-m_2+\binom{m_2}{v}\right)+\binom{m_1}{v}\binom{m_2}{v}2^{-\ell+1}(t-2v+1)(2v+1)\Big].$$

Proof. 1. As a first step we need to find the systematic form of the permuted parity check matrix, and the corresponding syndrome form. As a broad estimate we use the complexity of computing $U[H \mid s^\top]$, which takes approximately $(n-k)^2(n+1)$ bit operations.

2. To build the set S one has to compute $\pi_I(e_X)A_1^\top$ for all $e_X \in \mathbb{F}_2^n(X)$ of weight v. Using intermediate sums this costs
$$\ell(L(m_1,v)-m_1).$$

3. The set T is built similarly, since one has to compute $\pi_I(e_Y)A_1^\top + s_1$ for all $e_Y \in \mathbb{F}_2^n(Y)$ of weight v. Using intermediate sums this costs
$$\ell\left(L(m_2,v)-m_2+\binom{m_2}{v}\right),$$
where $\ell\binom{m_2}{v}$ is the complexity of adding s_1 to $\pi_I(e_Y)A_1^\top$ for each $e_Y \in \mathbb{F}_2^n(Y)$ of weight v.

4. In the next step we want to check for collisions between the set S and T. The set S consists of all e_X, where $e_X \in \mathbb{F}_2^n(X)$ has weight v. Hence S is of size $\binom{m_1}{v}$ and similarly the set T is of size $\binom{m_2}{v}$. The collision lives in $\mathbb{F}_2^\ell$, hence if we assume an uniform distribution we have to check on average
$$\frac{\binom{m_1}{v}\binom{m_2}{v}}{2^\ell}$$
many collisions. For each collision we have to compute $\pi_I(e_X+e_Y)A_2^\top + s_2$ and we only proceed if the weight of this is $t-2v$. With the method of early abort we only have to compute on average $2(t-2v+1)$ many entries. Each entry of the solution costs $2v+1$ bit operations.

5. This sums up to the cost of one iteration being

$$\begin{aligned} c(n,k,t,v,m_1,m_2,\ell) =& (n-k)^2(n+1)\\ &+\ell(L(m_1,v)-m_1)+\ell\left(L(m_2,v)-m_2+\binom{m_2}{v}\right)\\ &+\frac{\binom{m_1}{v}\binom{m_2}{v}}{2^\ell}2(t-2v+1)(2v+1). \end{aligned}$$

The success probability is given by having chosen the correct weight distribution of the error vector, i.e. in X the weight v, in Y the weight v, and in J the missing weight $t-2v$:

$$s(n,k,t,v,m_1,m_2,\ell)=\binom{m_1}{v}\binom{m_2}{v}\binom{n-k-\ell}{t-2v}\binom{n}{t}^{-1}.$$

Hence the overall cost of this algorithm is given as in the claim by

$$c(n,k,t,v,m_1,m_2,\ell)\cdot s(n,k,t,v,m_1,m_2,\ell)^{-1}.$$

□

3. Information set decoding over $\mathbb{Z}_4$

In this section we adapt our previous formulation of Lee-Brickell's and Stern's algorithm (in the collision formulation) for $\mathbb{Z}_4$-linear codes. For both algorithms, we first formulate the algorithm and illustrate how and why they work, before we determine their complexities.

Before explaining the algorithms, we determine the complexity of computing $Ax^\top$, for $x \in \mathbb{Z}_4^n$ having Lee weight w and a given matrix $A \in \mathbb{Z}_4^{k\times n}$, since this will be used in both algorithms.

Lemma 3.1. *Let $A \in \mathbb{Z}_4^{k\times n}$ and $x \in \mathbb{Z}_4^n$ with $wt_L(x) = w$. Then computing $Ax^\top$ needs at most $2(w-1)k$ bit operations.*

Proof. Whenever an entry of x is a 1, we need to add a column, and whenever it is a 3 we need to subtract a column. Both, a column addition or subtraction, cost at most $2k$ bit operations. If the entry of x is 2, we have to add a column twice; but the entry also has Lee weight 2, i.e., per Lee weight one we still get $2k$ operations. Thus, the overall complexity is $2(w-1)k$. □

3.1. Lee-Brickell's algorithm over $\mathbb{Z}_4$.

Recall that the systematic form of the parity check matrix is permutation equivalent to (2). For our purpose however, it is enough to consider the systematic form as

$$\begin{pmatrix} A & \mathrm{Id}_{n-k_1-k_2} \\ 2C & 0 \end{pmatrix},$$

where $A \in \mathbb{Z}_4^{(n-k_1-k_2)\times(k_1+k_2)}$ and $C \in \mathbb{Z}_2^{k_2\times(k_1+k_2)}$. The algorithm is given in Algorithm 3.

To illustrate the algorithm, let us again assume that the chosen information set is $\{1,\dots,k\}$. Then there exists an invertible $U \in \mathbb{Z}_4^{(n-k_1)\times(n-k_1)}$, such that

$$UHe^\top = \begin{pmatrix} A & \mathrm{Id}_{n-k_1-k_2} \\ 2C & 0 \end{pmatrix} \begin{pmatrix} e_1^\top \\ e_2^\top \end{pmatrix} = \begin{pmatrix} s_1^\top \\ 2s_2^\top \end{pmatrix} = Us^\top,$$

where $s_1 \in \mathbb{Z}_4^{n-k_1-k_2}, s_2 \in \mathbb{Z}_2^{k_2}$ and $e_1 \in \mathbb{Z}_4^{k_1+k_2}, e_2 \in \mathbb{Z}_4^{n-k_1-k_2}$. From this we get the conditions

$$\begin{aligned} e_1A^\top + e_2 &= s_1, \\ 2e_1C^\top &= 2s_2. \end{aligned}$$

We will choose e_1 having Lee weight w and such that the second condition is satisfied. The first condition will then be satisfied by the choice of e_2 and we check that e_2 has the remaining Lee weight $t-w$.

3.2. Complexity analysis of Lee-Brickell's algorithm over $\mathbb{Z}_4$.

We now estimate the complexity of Lee-Brickell's algorithm over $\mathbb{Z}_4$. We assume that one addition or one multiplication over $\mathbb{Z}_4$ costs 2 binary operations each. We remark here that if a lookup table for the multiplication and addition is used, the cost of one multiplication as well as the cost of one addition over $\mathbb{Z}_4$ will be only one bit. All the following costs, however, will be given without using a lookup table. Moreover, as in the binary case, we use school-book long multiplication instead of faster multiplication algorithms.

Algorithm 3 Lee-Brickell's Algorithm over $\mathbb{Z}_4$

Input: The $(n-k_1)\times n$ parity check matrix H over $\mathbb{Z}_4$, the syndrome $s\in\mathbb{Z}_4^{n-k_1}$ and the positive integers $t,w\in\mathbb{Z}$, such that $w\le 2(k_1+k_2)$, $w\le t$ and $t-w\le 2(n-k_1-k_2)$.
Output: $e\in\mathbb{Z}_4^n$ with $He^\top=s^\top$ and $wt_L(e)=t$.

1: Choose a quaternary information set $I\subset\{1,\dots,n\}$ of size k_1+k_2, let $J=\{1,\dots,n\}\setminus I$.
2: Find an invertible matrix $U\in\mathbb{Z}_4^{(n-k_1)\times(n-k_1)}$, such that

$$(UH)_I=\begin{pmatrix}A\\2C\end{pmatrix},\quad (UH)_J=\begin{pmatrix}\mathrm{Id}_{n-k_1-k_2}\\0\end{pmatrix},$$

where $A\in\mathbb{Z}_4^{(n-k_1-k_2)\times(k_1+k_2)}$ and $C\in\mathbb{Z}_2^{k_2\times(k_1+k_2)}$.
3: Compute $Us^\top=\begin{pmatrix}s_1^\top\\2s_2^\top\end{pmatrix}$, where $s_1\in\mathbb{Z}_4^{n-k_1-k_2}, s_2\in\mathbb{Z}_2^{k_2}$.
4: **for** $e_1\in\mathbb{Z}_4^n(I), wt_L(e_1)=w$ **do**
5: **if** $2\pi_I(e_1)C^\top=2s_2$ **then**
6: **if** $wt_L(s_1-\pi_I(e_1)A^\top)=t-w$ **then**
7: Output: $e=e_1+\sigma_J(s_1-\pi_I(e_1)A^\top)$
8: Start over with Step 1 and a new selection of I.

Theorem 3.2. *The average number of bit operations Algorithm 3 needs, is*

$$\frac{\binom{2n}{t}}{\binom{2(k_1+k_2)}{w}\binom{2(n-k_1-k_2)}{t-w}}\left[2(n-k_1)^2(n+1)+\binom{2(k_1+k_2)}{w}2(w(n-k_1)-k_2)\right].$$

Proof. As a first step we need to find the (permuted) quaternary systematic form of the parity check matrix, and the corresponding syndrome form. As a broad estimate we use the complexity of computing $U[H\mid s^\top]$, which takes approximately $(n-k_1)^2(n+1)$ quaternary operations, i.e., $2(n-k_1)^2(n+1)$ bit operations.

As a next step, we compute $2\pi_I(e_1)C^\top$ for all $e_1\in\mathbb{Z}_4^n(I)$ having $wt_L(e_1)=w$. With Lemma 3.1 this costs $2(w-1)k_2$ bit operations for one choice of e_1. Analogously, we get that computing $\pi_I(e_1)A^\top$ costs $2(w-1)(n-k_1-k_2)$ bit operations. Thus, computing $s_1-\pi_I(e_1)A^\top$ costs $2w(n-k_1-k_2)$ bit operations. Since there are $\binom{2(k_1+k_2)}{w}$ many such e_1 we get an overall cost of

$$\binom{2(k_1+k_2)}{w}(2w(n-k_1)-2k_2).$$

The success probability is given by having chosen the correct weight distribution of the error vector, i.e., weight t in the information set, and the missing weight $t-w$ in J:

$$\binom{2(k_1+k_2)}{w}\binom{2(n-k_1-k_2)}{t-w}\binom{2n}{t}^{-1}.$$

It follows that the overall cost of the algorithm is as claimed. □

3.3. COLLISION ISD (STERN'S ALGORITHM) OVER $\mathbb{Z}_4$. As in Lee-Brickell's algorithm, we first bring the parity check matrix into systematic form, according to the chosen information set. Because of the zero window of length ℓ we now split the matrix into three block rows, instead of two. If we assume that the information set

is $I = \{1, \dots, k_1 + k_2\}$ and the zero window is $\{k_1 + k_2 + 1, \dots, k_1 + k_2 + \ell\}$, we get the following situation:

$$UHe^\top = \begin{pmatrix} A & \mathrm{Id}_\ell & 0 \\ B & 0 & \mathrm{Id}_{n-k_1-k_2-\ell} \\ 2C & 0 & 0 \end{pmatrix} \begin{pmatrix} e_1^\top \\ 0 \\ e_2^\top \end{pmatrix} = \begin{pmatrix} s_1^\top \\ s_2^\top \\ 2s_3^\top \end{pmatrix} = Us^\top,$$

where $s_1 \in \mathbb{Z}_4^\ell, s_2 \in \mathbb{Z}_4^{n-k_1-k_2-\ell}, s_3 \in \mathbb{Z}_2^{k_2}$ and $A \in \mathbb{Z}_4^{\ell \times (k_1+k_2)}$, $B \in \mathbb{Z}_4^{(n-k_1-k_2-\ell)\times(k_1+k_2)}$, $C \in \mathbb{Z}_2^{k_2 \times (k_1+k_2)}$ and $e_1 \in \mathbb{Z}_4^{k_1+k_2}, e_2 \in \mathbb{Z}_4^{n-k_1-k_2-\ell}$. From this we get the conditions

$$\begin{aligned} e_1 A^\top &= s_1, \\ e_1 B^\top + e_2 &= s_2, \\ 2e_1 C^\top &= 2s_3. \end{aligned}$$

We will choose e_1 and e_2 having disjoint Lee weight $2v$ and $t - 2v$, respectively. In order to satisfy the first and the third condition, which only depend on e_1, we will check for a collision in the algorithm. The second condition will be satisfied by the choice of e_2. Thus, compared to the binary version we only get the extra conditions $2e_1C^\top = 2s_3$ on e_1. The rest is analogous. In fact we choose $e_1 = \pi_I(e_X + e_Y)$ and $e_2 = s_2 - e_1B^\top = s_2 - \pi_I(e_X + e_Y)B^\top$, where I is the quaternary information set, and X and Y are partitions of I. Therefore we get

$$UHe^\top = \begin{pmatrix} A\pi_I(e_X + e_Y)^\top \\ B\pi_I(e_X + e_Y)^\top + s_2^\top - B\pi_I(e_X + e_Y)^\top \\ 2C\pi_I(e_X + e_Y)^\top \end{pmatrix} = \begin{pmatrix} s_1^\top \\ s_2^\top \\ 2s_3^\top \end{pmatrix} = Us^\top.$$

The final collision algorithm is formulated in Algorithm 4. As in the binary case, we implicitly assume that we use intermediate sums in lines 6 and 7, early abort in line 10 and the speed up by using collisions in lines 8 and 9.

3.4. Complexity analysis of collision ISD over $\mathbb{Z}_4$. First we determine the complexities of the separate speed up concepts used in the main part of the algorithm.

1. Intermediate sums: The concept is the same over $\mathbb{Z}_4$ as over $\mathbb{F}_2$. With
$$\bar{L}(n, w) := \sum_{i=1}^{w} c(n, i) = \sum_{i=1}^{w} \binom{2n}{i}$$
and Lemma 3.1 we get that the cost of computing $Ax^\top$, for all $x \in \mathbb{Z}_4^n$ of Lee weight w, is $2k(\bar{L}(n, w) - 2n)$ bit operations.
2. Early abort: This concept changes slightly when using the Lee weight over $\mathbb{Z}_4$. On average we can expect $1/2$ of the entries to have weight 1, $1/4$ of the entries to have weight 2 and $1/4$ of the entries to have weight 0. Hence, on average, we should reach Lee weight t after $(\frac{1}{2} + 2\frac{1}{4})^{-1}t = t$ additions and therefore calculate $t + 1$ entries, before we can abort the computation.
3. Collisions: The average amount of collisions one needs to check between elements living in $\mathbb{Z}_4^n$ of a set S and a set T, under the assumption of a uniform distribution (analogously to the binary case), is given by
$$| S | \cdot | T | \cdot 4^{-n}.$$

Algorithm 4 Collision ISD (Stern's algorithm) over $\mathbb{Z}_4$

Input: The $(n-k_1)\times n$ parity check matrix H over $\mathbb{Z}_4$, the syndrome $s\in\mathbb{Z}_4^{n-k_1}$ and the non-negative integers $v, m_1, m_2, \ell \in \mathbb{Z}$, such that $k_1+k_2 = m_1+m_2$, $v \le 2m_1$, $v\le 2m_2$, $2v \le t$ and $t-2v \le 2(n-k_1-k_2-\ell)$.
Output: $e\in\mathbb{Z}_4^n$ with $He^\top = s^\top$ and $wt_L(e) = t$.

1: Choose a quaternary information set $I\subset\{1,\dots,n\}$ of size k_1+k_2.
2: Choose a set $Z\subset\{1,\dots,n\}\setminus I$, of size ℓ and define $J = \{1,\dots,n\}\setminus(I\cup Z)$.
3: Partition I into two disjoint sets X and Y of size m_1 and $m_2 = k_1+k_2-m_1$ respectively.
4: Find an invertible matrix $U\in\mathbb{Z}_4^{(n-k_1)\times(n-k_1)}$, such that

$$(UH)_I = \begin{pmatrix} A \\ B \\ 2C \end{pmatrix}, \quad (UH)_Z = \begin{pmatrix} \mathrm{Id}_\ell \\ 0 \\ 0 \end{pmatrix}, \quad (UH)_J = \begin{pmatrix} 0 \\ \mathrm{Id}_{n-k_1-k_2-\ell} \\ 0 \end{pmatrix},$$

where $A\in\mathbb{Z}_4^{\ell\times(k_1+k_2)}, B\in\mathbb{Z}_4^{(n-k_1-k_2-\ell)\times(k_1+k_2)}, C_1\in\mathbb{Z}_2^{k_2\times(k_1+k_2)}$.
5: Compute $Us^\top = \begin{pmatrix} s_1^\top \\ s_2^\top \\ 2s_3^\top \end{pmatrix}$, where $s_1\in\mathbb{Z}_4^\ell, s_2\in\mathbb{Z}_4^{n-k_1-k_2-\ell}$ and $s_3\in\mathbb{Z}_2^{k_2}$.
6: Compute the following set

$$S = \{(\pi_I(e_X)A^\top, 2\pi_I(e_X)C^\top, e_X)|\ e_X\in\mathbb{Z}_4^n(X), wt_L(e_X) = v\}.$$

7: Compute the following set

$$T = \{(s_1-\pi_I(e_Y)A^\top, 2s_3-2\pi_I(e_Y)C^\top, e_Y)|\ e_Y\in\mathbb{Z}_4^n(Y), wt_L(e_Y) = v\}.$$

8: **for** $(a,b,e_X)\in S$ **do**
9: **for** $(a,b,e_Y)\in T$ **do**
10: **if** $wt_L(s_2-\pi_I(e_X+e_Y)B^\top) = t-2v$ **then**
11: Output: $e = e_X+e_Y+\sigma_J(s_2-\pi_I(e_X+e_Y)B^\top)$
12: Start over with Step 1 and a new selection of I.

Theorem 3.3. *The average number of bit operations Algorithm 4 needs, is*

$$\begin{aligned}
&\frac{\binom{2n}{t}}{\binom{2m_1}{v}\binom{2m_2}{v}\binom{2(n-k_1-k_2-\ell)}{t-2v}}\Big[2(n-k_1)^2(n+1)\\
&+2\ell\left(\bar{L}(m_1,v)+\bar{L}(m_2,v)-2m_1-2m_2+\binom{2m_2}{v}\right)\\
&+k_2\left(L(m_1,v)+L(m_2,v)-m_1-m_2+2+\binom{2m_2}{v}\right)\\
&+\frac{c(m_1,v)c(m_2,v)}{2^{k_2+2\ell}}(t-2v+1)(4v-2)\Big],
\end{aligned}$$

assuming that $v\ge 1$.

Proof. 1. As a first step we need to find the (permuted) quaternary systematic form of the parity check matrix, and the corresponding syndrome form. As a broad estimate we use the complexity of computing $U[H\mid s^\top]$, which takes approximately $(n-k_1)^2(n+1)$ quaternary operations, i.e., $2(n-k_1)^2(n+1)$ bit operations.

2. To build the set S one has to compute $\pi_I(e_X)A^\top$ and $2\pi_I(e_X)C^\top$ for all $e_X \in \mathbb{Z}_4^n(X)$ of Lee weight v. Using intermediate sums the former costs $2\ell(\bar{L}(m_1, v) - 2m_1)$ and the latter costs $k_2(L(m_1, v) - m_1 + 1)$, since C lives in $\mathbb{Z}_2^{k_2 \times (k_1+k_2)}$ and $2\pi_I(e_X)$ lives in $2\mathbb{Z}_2^{k_1+k_2}$. Hence, computing S costs in total
$$2\ell(\bar{L}(m_1, v) - 2m_1) + k_2(L(m_1, v) - m_1 + 1)$$
binary operations.
3. The set T is built similarly, since one has to compute $s_1 - \pi_I(e_Y)A^\top$ and $2s_3 - 2\pi_I(e_Y)C^\top$ for all $e_Y \in \mathbb{Z}_4^n(Y)$ of weight v. Using intermediate sums we get a complexity of
$$2\ell\left(\bar{L}(m_2, v) - 2m_2 + \binom{2m_2}{v}\right) + k_2\left(L(m_2, v) - m_2 + 1 + \binom{m_2}{v}\right).$$
binary operations.
4. In the next step we want to check for the *two* collisions between the set S and T. The set S consists of all e_X, where $e_X \in \mathbb{Z}_4^n(X)$ has weight v. Hence S is of size $\binom{2m_1}{v}$ and similarly the set T is of size $\binom{2m_2}{v}$. The first collision lives in $\mathbb{Z}_4^\ell$, whereas the second collision lives in $2\mathbb{Z}_2^{k_2}$. We assume an uniform distribution and hence have to check on average
$$\frac{\binom{2m_1}{v}\binom{2m_2}{v}}{2^{k_2+2\ell}}$$
many collisions. For each collision we have to compute $s_2 - \pi_I(e_X + e_Y)B^\top$. With the method of early abort we only have to compute on average $t - 2v + 1$ entries. By Lemma 3.1, each entry of the solution costs $4v-2$ binary operations (assuming that $v \geq 1$).
5. This sums up to the cost of one iteration being
$$\begin{aligned} c(n, k_1, k_2, t, m_1, m_2, v, \ell) &:= 2(n-k_1)^2(n+1) \\ &\quad + 2\ell(\bar{L}(m_1, v) - 2m_1 + \bar{L}(m_2, v) - 2m_2) \\ &\quad + 2\ell\binom{2m_2}{v} + k_2\binom{2m_2}{v} + k_2(L(m_1, v) - m_1 + 1) \\ &\quad + k_2(L(m_2, v) - m_2 + 1) + \frac{\binom{2m_1}{v}\binom{2m_2}{v}}{2^{k_2+2\ell}}(t - 2v + 1)(4v - 2). \end{aligned}$$
The success probability is given by having chosen the correct weight distribution of the error vector, i.e. in X the weight v, in Y the weight v, and in J the missing weight $t - 2v$:
$$s(n, k_1, k_2, t, m_1, m_2, v, \ell) := \binom{2m_1}{v}\binom{2m_2}{v}\binom{2(n-k_1-k_2-\ell)}{t-2v}\binom{2n}{t}^{-1}.$$
Hence the overall cost of this algorithm is as in the claim given by
$$c(n, k_1, k_2, t, m_1, m_2, v, \ell) \cdot s(n, k_1, k_2, t, m_1, m_2, v, \ell)^{-1}.$$

□

4. Applications: Code-based cryptosystems over $\mathbb{Z}_4$

In this section we state a quaternary version of the McEliece and the Niederreiter cryptosystem. For the key generation one chooses a quaternary code $\mathcal{C}$ of length n and type $h = 4^{k_1}2^{k_2}$, which has an efficient decoding algorithm and is able to

correct up to t errors. We do not propose the use of a specific code, but we note that the secret code needs to come from a family of codes that is large enough and have a large enough error correction capacity t, such that brute force attacks on these aspects are not feasible.

Remark 2. We assume without loss of generality that the message x lives in $\mathbb{Z}_4^{k_1} \times \mathbb{Z}_2^{k_2}$. Indeed, we can transform any binary string $\bar{x} \in \mathbb{F}_2^{2k_1+k_2}$ into this form by an invertible map before the encryption, and use the inverse map after the decryption.

4.1. QUATERNARY MCELIECE. Let G be a $(k_1 + k_2) \times n$ generator matrix of $\mathcal{C}$ and choose an $n \times n$ permutation matrix P, this matrix has no further conditions, since the change of columns does not affect the $\mathbb{Z}_2$-part of the message, whereas for the $(k_1 + k_2) \times (k_1 + k_2)$ invertible matrix S, we need further conditions: in the classical case over finite fields, S is just a change of basis, but in the $\mathbb{Z}_4$ case, changing the rows of the generator matrix affects the position of the $\mathbb{Z}_2$-part of the message. Since such a change hinders the constructor of the cryptosystem to tell where the $\mathbb{Z}_2$-part of the message should be taken, we will restrict the choice of invertible matrices to the following form: let S_1 and S_2 be $k_1 \times k_1$, respectively $k_2 \times k_2$ invertible matrices over $\mathbb{Z}_4$, then S is given by

$$S = \begin{pmatrix} S_1 & 0 \\ 0 & S_2 \end{pmatrix}.$$

Compute $G' = SGP$ and publish (k_1, k_2, G', t).

For the encryption, let $x = (x_1, x_2)$, with $x_1 \in \mathbb{Z}_4^{k_1}$ and $x_2 \in \mathbb{Z}_2^{k_2}$ be the message and choose an error vector $e \in \mathbb{Z}_4^n$ of Lee weight $\mathrm{wt}_L(e) \leq t$. The cipher is computed as

$$y = xG' + e.$$

For the decryption one computes

$$yP^{-1} = xSG + eP^{-1}.$$

Since $\mathrm{wt}_L(eP^{-1}) \leq t$ and SG generates the same code as G we can use the decoding algorithm of the code to recover xS and hence the message x.

Remark 3. To outgo a chosen ciphertext attack (CCA) one can multiply a new permutation matrix to the public generator matrix at each new instantiation of the system, analogously to the classical McEliece system [5].

4.2. QUATERNARY NIEDERREITER. The quaternary version of the Niederreiter cryptosystem is done in a similar way by using the parity check matrix H and by computing its syndromes for encryption. Since there is no restriction on the message space in the Niederreiter version, there will be no conditions needed on the permutation matrix and on the invertible matrix.

Again, one chooses a quaternary code $\mathcal{C}$ of length n and type $h = 4^{k_1}2^{k_2}$, which has an efficient decoding algorithm and is able to correct up to t errors.

Let H be a $(n-k_1) \times n$ parity matrix of $\mathcal{C}$, choose an invertible $(n-k_1) \times (n-k_1)$ matrix S, i.e. $\det(S) \in \mathbb{Z}_4^\times$ and an $n \times n$ permutation matrix P.[2] Compute $H' = S^{-1}HP$ and publish (k_1, k_2, H', t).

For the encryption, let $x \in \mathbb{Z}_4^n$ be the message of Lee weight $\mathrm{wt}_L(x) \leq t$. The cipher is computed as

$$y^\top = H'x^\top.$$

[2] Also here we can choose a new P every time to prevent a CCA.

For the decryption one computes

$$Sy^\top = HPx^\top.$$

Since $\mathrm{wt}_L(Px^\top) \leq t$ we can use the decoding algorithm of the code to recover $Px^\top$ and hence the message x.

4.3. Key size. To determine the key size we need to count the number of non-prescribed entries of the public generator matrix. For this we assume that the generator matrix is published in quaternary systematic form as in (1).

This allows us to compute the size of the generator matrix in the form (1), or equivalently the size of the parity check matrix in the form (2).

Theorem 4.1. *The size of the public key, given by the non-prescribed parts of either the generator matrix (1) or the parity check matrix (2), is*

$$2(n-k_1-k_2)k_1 + (n-k_1-k_2)k_2 + k_1k_2 = k_1k_2 + (2k_1+k_2)(n-k_1-k_2)$$

bits.

In the following we study the key sizes of the proposed cryptographic scheme in Section 4 with respect to a given security level against Algorithm 4 provided in Section 2.2.

Two of the most studied families of $\mathbb{Z}_4$-linear codes are Kerdock and Preparata codes [21]. Because of their small minimum distance Preparata codes are not useful for our cryptosystems. Even though Kerdock codes over $\mathbb{Z}_4$ satisfy all the conditions needed for the quaternary version of the McEliece cryptosystem, they seem to be a bad choice for key size reasons: while the key size of the cryptosystems doubles going from code length $n = 2^m$ to 2^{m+1}, the security level only increases by 3 bits.

For now we leave it as an open problem to find suitable codes for the use in a Lee-metric public key cryptosystem, but we remark that many constructions of Lee codes are known, e.g., [7, 10, 17, 21, 22, 25, 36, 38, 43, 47]. For the remainder of this paper we will use only theoretical parameters, to illustrate how using the Lee metric could potentially decrease the key sizes in a McEliece or Niederreiter type cryptosystem. We consider quaternary codes achieving the Gilbert-Varshamov bound in the Lee metric, i.e., codes of length n and Lee weight d whose cardinality is at least

$$\frac{4^n}{(\sum_{j=0}^{d-1}\binom{2n}{j}-1)3+1}.$$

Example 4.2. As a first example we examine codes of length $n = 150$ and minimum Lee distance $d = 81$, i.e., we can set $t = 40$. The Gilbert-Varshamov bound tells us that such codes with $\mathbb{Z}_4$-dimension $26 = k_1 + k_2/2$ exist. We now vary k_1 from 1 to 25, with $k_2 = 2(26-k_1)$. Furthermore, we set $m_1 = \lceil (k_1+k_2)/2 \rceil, m_2 = \lfloor (k_1+k_2)/2 \rfloor$ and we optimize on the size ℓ of the zero-window and the number $2v$ of errors in the information set. With these parameters we get the following key sizes and security levels in bits, see Table 1.

Note that the above security levels were computed using Algorithm 4, which always outperforms Algorithm 3 on a classical computer. For comparison, a binary code of length $2n = 300$, dimension $k = 26$ and minimum Hamming distance $d = 81$ gives a key size of 7124 and a security level of 27 bits with the binary version of

k_1	1	2	3	...	18	19	...	24	25	26
best ℓ	0	0	0	...	0	0	...	0	1	2
best v	4	4	4	...	3	3	...	2	2	2
key size	5198	5296	5390	...	6110	6160	...	6440	6446	6448
security level	31	31	31	...	27	27	...	28	28	28

TABLE 1. Key sizes and security levels (both in bits) for GV-codes over $\mathbb{Z}_4$ with $n = 150$ and $d = 81$.

Stern's algorithm. Hence, depending on k_1 we get at least the same security level with a key size improvement of around $10 - 28\%$, when using Lee codes over $\mathbb{Z}_4$.[3]

In the next example we find codes that theoretically achieve a security level of 128 bits, against the adaptation of the collision ISD algorithm.

Example 4.3. Given the relative distance $d/n = 0.2$, using an optimization on the size of v and ℓ we search for a minimal code length n, such that a k_1 exists with which the security level of 128 bits is reached. We get $n = 425, k = 229, k_1 = 33, k_2 = 392, t = 42, v = 21, \ell = 0$ and key size of 12936 bits.

Remark 4. The previously obtained theoretical values give much smaller public keys than the classical McEliece system with binary Goppa codes achieves. For this note that for the security level of 128 bits, the proposed parameters for the McEliece system using Goppa codes by Bernstein *et al.* in [5] are $n = 2960, k = 2288$, which gives a key size of 1537536 bits. In fact the theoretical key sizes presented here are within the range of quasi-cyclic MDPC codes, which were submitted to NIST for post-quantum code-based cryptosystem (from 10 to 37 kilobits, see https://csrc.nist.gov/projects/post-quantum-cryptography/round-2-submissions).

5. Generalization from $\mathbb{Z}_4$ to $\mathbb{Z}_{p^s}$

In this section we give the general idea of how to generalize the ISD algorithm and the two code-based cryptosystems to any Galois ring $\mathbb{Z}_{p^s}$ for any prime p and $s \in \mathbb{N}$. Recall that the Lee weight of $x \in \mathbb{Z}_{p^s}^n$ is given by

$$\mathrm{wt}_L(x) = \sum_{i=0}^{n} \min\{x_i, p^s - x_i\}.$$

The main modification is in the systematic form of the generator and parity check matrix of the code. In general, a linear code over $\mathbb{Z}_{p^s}$ has a (column permuted) generator matrix of the form

$$G = \begin{pmatrix} \mathrm{Id}_{k_1} & A_{1,2} & A_{1,3} & \dots & A_{1,s} & A_{1,s+1} \\ 0 & p\,\mathrm{Id}_{k_2} & pA_{2,3} & \dots & pA_{2,s} & pA_{2,s+1} \\ 0 & 0 & p^2\,\mathrm{Id}_{k_3} & \dots & p^2A_{3,s} & p^2A_{3,s+1} \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & p^{s-1}\,\mathrm{Id}_{k_s} & p^{s-1}A_{s,s+1} \end{pmatrix}, \tag{4}$$

[3]For further comparison, binary codes achieving the Gilbert-Varshamov bound of length $2n = 300$ and dimension $k = 26$ have minimum distance $d = 102$ and achieve a security level of 28 bits with a key size of 7124 bits.

and a (column permuted) parity check matrix of the form

$$H = \begin{pmatrix} B_{1,1} & B_{1,2} & \dots & B_{1,s-1} & B_{1,s} & \mathrm{Id}_{n-K} \\ pB_{2,1} & pB_{2,2} & \dots & pB_{2,s-1} & p\,\mathrm{Id}_{k_s} & 0 \\ p^2B_{3,1} & p^2B_{3,2} & \dots & p^2\,\mathrm{Id}_{k_{s-1}} & 0 & 0 \\ \vdots & \vdots & \ddots & & \vdots & \vdots \\ p^{s-1}B_{s,1} & p^{s-1}\,\mathrm{Id}_{k_2} & \dots & 0 & 0 & 0 \end{pmatrix}, \tag{5}$$

where $K = \sum_{i=1}^{s} k_i$ and the matrices live in

$$\text{for } j \le s : A_{i,j} \in \mathbb{Z}_{p^{s+1-i}}^{k_i \times k_j}, \quad \text{and} \quad A_{i,s+1} \in \mathbb{Z}_{p^{s+1-i}}^{k_i \times (n-K)}$$
$$\text{for } i > 1 : B_{i,j} \in \mathbb{Z}_{p^{s+1-i}}^{k_{s-j+2} \times k_j}, \quad \text{and} \quad B_{1,j} \in \mathbb{Z}_{p^{s+1-i}}^{(n-K) \times k_j}.$$

We say that such a code has type $(p^s)^{k_1}(p^{s-1})^{k_2} \dots p^{k_s}$. This is also the cardinality of the code, and the uniquely encodable messages are of the form $(m_1, m_2, \dots, m_s) \in \mathbb{Z}_{p^s}^{k_1} \times \mathbb{Z}_{p^{s-1}}^{k_2} \times \dots \times \mathbb{Z}_{p}^{k_s}$. If our code has length n, an information set is a set $I \subseteq \{1, \dots, n\}$ of size K such that $|\mathcal{C}_I| = |\mathcal{C}|$.

5.1. INFORMATION SET DECODING OVER $\mathbb{Z}_{p^s}$. One can set up an ISD algorithm analogously to Algorithm 4. Instead of three conditions on e_1 and e_2 we then get $s+1$ conditions on e_i for $i \in \{1, \dots s\}$, where e_s is only part of one condition. For e_i with $i \in \{1, \dots s-1\}$ to satisfy the s conditions, that are not involving e_s, one needs to compute similar sets S_i for $i \in \{1, \dots, s-1\}$ consisting of s tuples, find the collisions between them, and lastly choose e_s satisfying the remaining condition. In the appendix we exemplify the described algorithm over $\mathbb{Z}_8$. Note that this ISD algorithm is different to the Lee metric ISD algorithm proposed in [46][4], as there the systematic form is considered to be

$$H = \begin{pmatrix} A & \mathrm{Id}_{n-K} \\ pB & 0 \end{pmatrix},$$

with $A \in \mathbb{Z}_{p^s}^{(n-K)\times K}$ and $B \in \mathbb{Z}_{p^{s-1}}^{(K-k_1)\times K}$. This choice of systematic form clearly makes the ISD algorithm easier to understand, but does not take into account the particular form of the parity check matrix. We leave it as an open problem to compare the benefits of the two different algorithms.

5.2. CODE-BASED CRYPTOSYSTEMS OVER $\mathbb{Z}_{p^s}$. The Niederreiter system does not need a modification of 4.2 to be used over any $\mathbb{Z}_{p^s}$. In the McEliece cryptosystem 4.1 one just needs to make sure that the invertible matrix S has the correct block diagonal structure to prevent mixing the subcodes that live in different subrings. The rest stays the same.

For the size of the public key note that you can either publish the generator or the parity check matrix. However, it turns out that both ways give you the same key size. The key size in bits related to the generator matrix G as in (4) (or equivalently related to the parity check matrix H as in (5)) is

$$\sum_{i=1}^{s}\Big(\sum_{j=i+1}^{s} k_i k_j \log_2(p^{s+1-i}) + k_i(n-K)\log_2(p^{s+1-i})\Big)$$
$$= \sum_{i=1}^{s} k_i \log_2(p^{s+1-i}) \sum_{j=i+1}^{s} (k_j + n - K).$$

[4] [46] appeared as a follow-up of this work.

6. Conclusion

The change from the Hamming metric to the rank metric has recently received a lot of attention in the code-based cryptography community, since the key sizes are very promising. Following this idea, we propose the change to the Lee metric and the ring-linear codes related to this metric. In this paper we built the framework for the use of quaternary codes in code-based cryptography by generalizing Lee-Brickell's and Stern's ISD algorithm to $\mathbb{Z}_4$. This paper also gives the general form of the quaternary version of the McEliece and the Niederreiter cryptosystem.

Here we provide some questions, which might lead to interesting applications and further understanding of ring-linear codes and the Lee metric from a cryptographic point of view. Even though we restricted the focus in this paper to the case $\mathbb{Z}_4$, and explained shortly how to generalize this to $\mathbb{Z}_{p^s}$, it is possible and it might be interesting to generalize this to $\mathbb{Z}_m$, for any m. It is possible that some of the ISD algorithms have a structure that correlates better to the Lee metric, hence there might be other ISD algorithms which should be generalized to $\mathbb{Z}_4$. And the most important question in order to have an application in cryptography is: which codes might be used for the quaternary version of the McEliece cryptosystem, such that the conditions for the cryptosystem are satisfied and the key size is reasonable?

Acknowledgments

We would like to thank Karan Khathuria for fruitful discussions and technical support.

References

[1] E. F. Assmus and H. F. Mattson, Error-correcting codes: An axiomatic approach, *Information and Control*, **6** (1963), 315–330.
[2] A. Becker, A. Joux, A. May and A. Meurer, Decoding random binary linear codes in $2^{n/20}$: How 1+ 1= 0 improves information set decoding, *Advances in Cryptology–EUROCRYPT 2012, Lecture Notes in Comput. Sci.*, Springer, Heidelberg, **7237** (2012), 520–536.
[3] E. Berlekamp, *Algebraic Coding Theory*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2015.
[4] D. J. Bernstein, Grover vs. McEliece, In *International Workshop on Post-Quantum Cryptography*, *Lecture Notes in Comput. Sci.,* Springer, **6061** (2010), 73–80.
[5] D. J. Bernstein, T. Lange and C. Peters, Attacking and defending the McEliece cryptosystem, In *International Workshop on Post-Quantum Cryptography*, Springer, (2008), 31–46.
[6] D. J. Bernstein, T. Lange and C. Peters, Smaller decoding exponents: Ball-collision decoding, In *Annual Cryptology Conference*, Springer, (2011), 743–760.
[7] T. Blackford, Cyclic codes over $\mathbb{Z}_4$ of oddly even length, *Discrete Applied Mathematics*, **128** (2003), 27–46.
[8] I. F. Blake, Codes over certain rings, *Information and Control*, **20** (1972), 396–404.
[9] I. F. Blake, Codes over integer residue rings, *Information and Control*, **29** (1975), 295–300.
[10] E. Byrne, Decoding a class of Lee metric codes over a Galois ring, *IEEE Transactions on Information Theory*, **48** (2002), 966–975.
[11] A. Canteaut and Hervé Chabanne, *A Further Improvement of the Work Factor in an Attempt at Breaking McEliece's Cryptosystem*, PhD thesis, INRIA, 1994.
[12] A. Canteaut and F. Chabaud, A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511, *IEEE Transactions on Information Theory*, **44** (1998), 367–378.
[13] A. Canteaut and N. Sendrier, Cryptanalysis of the original McEliece cryptosystem, In *Advances in Cryptology ASIACRYPT'98 (Beijing), Lecture Notes in Comput. Sci.,* Springer, Berlin, **1514** (1998), 187–199.

[14] F. Chabaud, Asymptotic analysis of probabilistic algorithms for finding short codewords, *Eurocode '92 (Udine, 1992), CISM Courses and Lect.,* Springer, Vienna, **339** (1993), 175–183.
[15] J. T. Coffey and R. M. Goodman, The complexity of information set decoding, *IEEE Transactions on Information Theory*, **36** (1990), 1031–1037.
[16] I. I. Dumer, Two decoding algorithms for linear codes, *Problemy Peredachi Informatsii*, **25** (1989), 24–32.
[17] T. Etzion, A. Vardy and E. Yaakobi, Dense error-correcting codes in the Lee metric, In *IEEE Information Theory Workshop*, (2010), 1–5.
[18] M. Finiasz and N. Sendrier, Security bounds for the design of code-based cryptosystems, In *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, (2009), 88–105.
[19] E. Gabidulin, A brief survey of metrics in coding theory, *Mathematics of Distances and Applications*, **66** (2012).
[20] M. Greferath, An introduction to ring-linear coding theory, In *Gröbner Bases, Coding, and Cryptography*, Springer, (2009), 219–238.
[21] R. A. Hammons, V. P. Kumar, R. A. Calderbank, N. Sloane and P. Solé, The $\mathbb{Z}_4$-linearity of Kerdock, Preparata, Goethals, and related codes, *IEEE Transactions on Information Theory*, **40** (1994), 301–319.
[22] T. Helleseth and V. Zinoviev, On $\mathbb{Z}_4$-linear Goethals codes and Kloosterman sums, *Designs, Codes and Cryptography*, **17** (1999), 269–288.
[23] S. Hirose, May-Ozerov algorithm for nearest-neighbor problem over $\mathbb{F}_q$ and its application to information set decoding, In *International Conference for Information Technology and Communications*, Springer, (2016), 115–126.
[24] C. Interlando, K. Khathuria, N. Rohrer, J. Rosenthal and V. Weger, Generalization of the Ball-Collision algorithm, preprint, `arXiv::`1812.10955, 2018.
[25] D. S. Krotov, $\mathbb{Z}_4$-linear Hadamard and extended perfect codes, *Electron. Notes Discrete Math., Elsevier Sci. B. V., Amsterdam*, **6** (2001), 107–112.
[26] E. A. Kruk, Decoding complexity bound for linear block codes, *Problemy Peredachi Informatsii*, **25** (1989), 103–107.
[27] P. J. Lee and E. F. Brickell, An observation on the security of McEliece's public-key cryptosystem, In *Workshop on the Theory and Application of of Cryptographic Techniques*, Springer, **330** (1988), 275–280.
[28] J. S. Leon, A probabilistic algorithm for computing minimum weights of large error-correcting codes, *IEEE Transactions on Information Theory*, **34** (1988), 1354–1359.
[29] A. May, A. Meurer and E. Thomae, Decoding random linear codes in $\mathcal{O}(2^{0.054n})$, In *International Conference on the Theory and Application of Cryptology and Information Security*, Springer, **2011** (2011), 107–124.
[30] A. May and I. Ozerov, On computing nearest neighbors with applications to decoding of binary linear codes, In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, (2015), 203–228.
[31] R. J. McEliece, *A Public-Key Cryptosystem Based on Algebraic Coding Theory*, Technical report, DSN Progress report, Jet Propulsion Laboratory, Pasadena, 1978.
[32] A. Meurer, *A Coding-Theoretic Approach to Cryptanalysis*, PhD thesis, Ruhr University Bochum, 2012.
[33] A. A. Nechaev, Kerdock code in a cyclic form, *Discrete Mathematics and Applications*, **1** (1991), 365–384.
[34] R. Niebuhr, E. Persichetti, Pierre-Louis Cayrel, S. Bulygin and J. Buchmann, On lower bounds for information set decoding over $\mathbb{F}_q$ and on the effect of partial knowledge, *International journal of information and Coding Theory*, **4** (2017), 47–78.
[35] C. Peters, Information-set decoding for linear codes over $\mathbb{F}_q$, In *International Workshop on Post-Quantum Cryptography*, Springer, **6061** (2010), 81–94.
[36] V. S. Pless and Z. Qian, Cyclic codes and quadratic residue codes over $\mathbb{Z}_4$, *IEEE Transactions on Information Theory*, **42** (1996), 1594–1600.
[37] E. Prange, The use of information sets in decoding cyclic codes, *IRE Transactions on Information Theory*, **8** (1962), 5–9.
[38] R. M. Roth and P. H. Siegel, Lee-metric BCH codes and their application to constrained and partial-response channels, *IEEE Transactions on Information Theory*, **40** (1994), 1083–1096.

[39] C. Satyanarayana, Lee metric codes over integer residue rings (corresp), *IEEE Transactions on Information Theory*, **25** (1979), 250–254.
[40] P. Shankar, On BCH codes over arbitrary integer rings (corresp), *IEEE Transactions on Information Theory*, **25** (1979), 480–483.
[41] E. Spiegel, Codes over $\mathbb{Z}_m$, *Information and control*, **35** (1977), 48–51.
[42] J. Stern, A method for finding codewords of small weight, In *International Colloquium on Coding Theory and Applications*, Springer, **388** (1989), 106–113.
[43] I. Tal and R. M. Roth, On list decoding of alternant codes in the Hamming and Lee metrics, In *IEEE International Symposium on Information Theory*, 2003, 364–364.
[44] H. Tapia-Recillas, A secret sharing scheme from a chain ring linear code, *Congressus Numerantium*, **186** (2007), 33–39.
[45] J. van Tilburg, On the McEliece public-key cryptosystem, In *Conference on the Theory and Application of Cryptography*, Springer, **403** (1990), 119–131.
[46] V. Weger, M. Battaglioni, P. Santini, F. Chiaraluce, M. Baldi and E. Persichetti, Information set decoding of Lee-metric codes over finite rings, arXiv preprint, arXiv:2001.08425, 2020.
[47] Y. Wu and C. N. Hadjicostis, Decoding algorithm and architecture for BCH codes under the Lee metric, *IEEE Transactions on Communications*, **56** (2008), 2050–2059.

APPENDIX

Here we describe the ISD algorithm from Section 5 over $\mathbb{Z}_8$. We consider a code $\mathcal{C} \subseteq \mathbb{Z}_8^n$ of type $|\mathcal{C}| = 8^{k_1}4^{k_2}2^{k_3}$, and define $K := k_1 + k_2 + k_3$. For simplicity let us assume that the information set is $I = \{1, \dots, K\}$ with $I_1 = \{1, \dots, k_1 + k_2\}$, $I_2 = \{k_1 + k_2 + 1, \dots, K\}$ and the zero window is $\{K+1, \dots, K+\ell\}$. We first bring the parity check matrix into systematic form, according to the chosen information set, and get:

$$UHe^\top = \begin{pmatrix} A & B & \mathrm{Id}_\ell & 0 \\ C & D & 0 & \mathrm{Id}_{n-K-\ell} \\ 2E & 2\mathrm{Id}_{k_3} & 0 & 0 \\ 4F & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} e_1^\top \\ e_2^\top \\ 0 \\ e_3^\top \end{pmatrix} = \begin{pmatrix} s_1^\top \\ s_2^\top \\ 2s_3^\top \\ 4s_4^\top \end{pmatrix} = Us,$$

where $s_1 \in \mathbb{Z}_8^\ell, s_2 \in \mathbb{Z}_8^{n-K-\ell}, s_3 \in \mathbb{Z}_4^{k_3}, s_4 \in \mathbb{Z}_2^{k_2}$ and $A \in \mathbb{Z}_8^{\ell \times (k_1+k_2)}$, $B \in \mathbb{Z}_8^{\ell \times k_3}$, $C \in \mathbb{Z}_8^{(n-K-\ell)\times(k_1+k_2)}, D \in \mathbb{Z}_8^{(n-K-\ell)\times k_3}, E \in \mathbb{Z}_4^{k_3 \times (k_1+k_2)}, F \in \mathbb{Z}_2^{k_2 \times (k_1+k_2)}$ and $e_1 \in \mathbb{Z}_8^{k_1+k_2}, e_2 \in \mathbb{Z}_8^{k_3}, e_3 \in \mathbb{Z}_8^{n-K-\ell}$. From this we get the conditions

$$\begin{aligned} e_1A^\top + e_2B^\top &= s_1, \\ e_1C^\top + e_2D^\top + e_3 &= s_2, \\ 2e_1E^\top + 2e_2 &= 2s_3, \\ 4e_1F^\top &= 4s_4. \end{aligned}$$

We will choose e_1 and e_2 disjoint both having Lee weight v and e_3 having Lee weight $t - 2v$. In order to satisfy the first, the third and the fourth condition, which only depend on e_1 and e_2, we will check for a collision within the algorithm. The second condition will be satisfied by choosing $e_3 = s_2 - e_1C^\top - e_2D^\top$. The algorithm is provided in Algorithm 5.

Received May 2019; revised February 2020.

E-mail address: anna-lena.horlemann@unisg.ch
E-mail address: violetta.weger@math.uzh.ch

Algorithm 5 Collision ISD (Stern's algorithm) over $\mathbb{Z}_8$

Input: The $(n-k_1)\times n$ parity check matrix H over $\mathbb{Z}_8$, the syndrome $s\in\mathbb{Z}_8^{n-k_1}$ and the non-negative integers $v,\ell\in\mathbb{Z}$, such that $v\leq\min\{4(k_1+k_2),4k_3\}$, $2v\leq t$ and $t-2v\leq 4(n-K-\ell)$.
Output: $e\in\mathbb{Z}_8^n$ with $He^\top=s^\top$ and $wt_L(e)=t$.

1: Choose a quaternary information set $I\subset\{1,\dots,n\}$ of size $K=k_1+k_2+k_3$ with the corresponding subsets I_1 of size (k_1+k_2) and I_2 of size k_3.
2: Choose a set $Z\subset\{1,\dots,n\}\setminus I$, of size ℓ and define $J=\{1,\dots,n\}\setminus(I\cup Z)$.
3: Find an invertible matrix $U\in\mathbb{Z}_8^{(n-k_1)\times(n-k_1)}$, such that

$$(UH)_{I_1}=\begin{pmatrix}A\\C\\2E\\4F\end{pmatrix},\qquad (UH)_{I_2}=\begin{pmatrix}B\\D\\2\,\mathrm{Id}_{k_3}\\0\end{pmatrix},$$
$$(UH)_Z=\begin{pmatrix}\mathrm{Id}_\ell\\0\\0\\0\end{pmatrix},\qquad (UH)_J=\begin{pmatrix}0\\\mathrm{Id}_{n-K-\ell}\\0\\0\end{pmatrix},$$

where $A\in\mathbb{Z}_8^{\ell\times(k_1+k_2)}$, $B\in\mathbb{Z}_8^{\ell\times k_3}$, $C\in\mathbb{Z}_8^{(n-K-\ell)\times(k_1+k_2)}$, $D\in\mathbb{Z}_8^{(n-K-\ell)\times k_3}$, $E\in\mathbb{Z}_4^{k_3\times(k_1+k_2)}$ and $F\in\mathbb{Z}_2^{k_2\times(k_1+k_2)}$.
4: Compute $Us^\top=\begin{pmatrix}s_1^\top\\s_2^\top\\2s_3^\top\\4s_4^\top\end{pmatrix}$, where $s_1\in\mathbb{Z}_8^\ell, s_2\in\mathbb{Z}_8^{n-K-\ell}, s_3\in\mathbb{Z}_4^{k_3}$ and $s_4\in\mathbb{Z}_2^{k_2}$.
5: Compute the following set

$$S=\{(e_1A^\top,2e_1E^\top,4e_1F^\top,e_1)|\ e_1\in\mathbb{Z}_8^{k_1+k_2},wt_L(e_1)=v\}.$$

6: Compute the following set

$$T=\{(s_1-e_2B^\top,2s_3-2e_2,4s_4,e_2)|\ e_2\in\mathbb{Z}_8^{k_3},wt_L(e_2)=v\}.$$

7: **for** $(a,b,c,e_1)\in S$ **do**
8: **for** $(a,b,c,e_2)\in T$ **do**
9: **if** $wt_L(s_2-e_1C^\top-e_2D^\top)=t-2v$ **then**
10: Output: $e_{I_1}=e_1,e_{I_2}=e_2,e_Z=0$ and $e_J=s_2-e_1C^\top-e_2D^\top$.
11: Start over with Step 1 and a new selection of I.