J. Algebra Comb. Discrete Appl.
7(2) • 195–207

Received: 10 September 2019
Accepted: 18 April 2020

Journal of Algebra Combinatorics Discrete Structures and Applications

Generalization of the ball-collision algorithm

Research Article

Carmelo Interlando, Karan Khathuria, Nicole Rohrer, Joachim Rosenthal, Violetta Weger

Abstract: In this paper we generalize the ball-collision algorithm by Bernstein, Lange, Peters from the binary field to a general finite field. We also provide a complexity analysis and compare the asymptotic complexity to other generalized information set decoding algorithms.

2010 MSC: 94B35, 94A60

Keywords: Coding theory, ISD, Ball-collision

1. Introduction

Since 1978 it has been known that decoding a random linear code is an NP-complete problem, this was shown in [7] by Berlekamp, McEliece and van Tilborg. Therefore the interesting task arises of finding the complexity of decoding a random linear code using the best algorithms available. Until today two main methods for decoding have been proposed: information set decoding (ISD) and the generalized birthday algorithm (GBA). The ISD is more efficient if the decoding problem has only a small number of solutions, whereas GBA is efficient when there are many solutions. Also other ideas such as statistical decoding [1], gradient decoding [2] and supercode decoding [5] have been proposed but fail to outperform ISD algorithms. The input of an ISD algorithm is a corrupted codeword and it outputs the error vector. ISD 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 are based on a decoding algorithm proposed by Prange [28] in 1962 and their structures do not change much from the original: as a first step one chooses an information set, then Gaussian elimination brings the parity check matrix in a standard form and assuming that the errors are outside of the information set, these row operations on the syndrome will exploit the error vector, if the weight does not exceed the given error correction capacity.

Carmelo Interlando; Department of Mathematics and Statistics, San Diego State University, San Diego, CA 92182-7720 (email: carmelo.interlando@sdsu.edu).
Karan Khathuria (Corresponding Author), Nicole Rohrer, Joachim Rosenthal, Violetta Weger; Institute of Mathematics, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland (email: karan.khathuria@math.uzh.ch, nicole.rohrer@uzh.ch, rosenthal@math.uzh.ch, violetta.weger@math.uzh.ch).

The problem of decoding random linear codes has recently been receiving prominence with the proposal of using code-based public key cryptosystems for an upcoming post-quantum cryptographic public key standard. The idea of using linear codes in public key cryptography was first formulated by Robert McEliece [24]. Since the publication of McEliece a large amount of research has been done and the interested reader will find more information in a recent survey [9].

If the secret code is hidden well enough an adversary who wants to break a code-based cryptosystem encounters the decoding problem of a random linear code. It is therefore of crucial importance to understand the complexity of the best algorithms capable of decoding a general linear code.

The ISD algorithms were often considered when proposing a variant of the McEliece cryptosystem, to find the key size needed for a given security level. ISD algorithms hence do not break a code-based cryptosystem but they determine the choice of secure parameters. Since some of the new proposals (for example [3, 4, 18]) involve codes over general finite fields, having efficient ISD algorithms generalized to $\mathbb{F}_q$ is an essential problem.

Bernstein, Lange and Peters found a clever improvement of the ISD algorithm which they called ball-collision decoding [8]. The algorithm of Bernstein et. al. was presented for random binary linear codes. The main contribution of our paper is a generalization of the ball-collision decoding algorithm to arbitrary finite fields.

The paper is structured as follows: in Section 2 we discuss the previous work on ISD algorithms focusing on those which have been generalized to an arbitrary finite field. In Section 3 we describe the ball-collision algorithm over the binary field and the notations and concepts involved in the algorithm. In Section 4 we present the ball-collision algorithm over $\mathbb{F}_q$ and in Section 5 we perform the complexity analysis of our algorithm including numerical parameter optimization and asymptotic analysis.

2. Related work

Eventhough the cost of one iteration of Prange's original ISD algorithm was very low, the algorithm was still coming with a huge complexity due to the number of iterations needed. Many improvements have been suggested to Prange's simplest form of ISD (see for example [11–13, 19, 21, 31]), they all focus on a more elaborate and more likely weight distribution of the error vector, which results in a higher cost of one iteration, but less iterations have to be performed. The improvements were splitting from an early time on into two directions: the first direction is following the splitting of Lee and Brickell [20] into the information set and the redundant set, i.e. they ask for v errors in the information set and $t-v$ outside. The second direction is Dumer's splitting approach [13], which is asking for v errors in $k+\ell$ bits, which are containing an information set, and $t-v$ in the remaining $n-k-\ell$ bits. Apart from improvements regarding the success probability, one can also improve the cost of one iteration: Canteaut and Chabaud [10] have provided a speed up for finding information sets. They show that the information set should not be taken at random after one unsuccessful iteration, but rather a part of the previous information set should be reused and therefore a part of the Gaussian elimination step is already performed.

The second direction has resulted in many improvements, for example in 2009 Finiasz and Sendrier [14] have built two intersecting subsets of the $k+\ell$ bits, which contain an information set, and ask for v disjoint errors in both sets and $t-2v$ in the remaining $n-k-\ell$ bits. Niebuhr, Persichetti, Cayrel, Bulygin and Buchmann [26] in 2010 improved the performance of ISD algorithms over $\mathbb{F}_q$ based on the idea of Finiasz and Sendrier [14]. In 2011 May, Meurer and Thomae [22] proposed an improvement using the representation technique introduced by Howgrave-Graham and Joux [17]. To this algorithm Becker, Joux, May and Meurer [6] (BJMM) in 2012 introduced further improvements. In the same year Meurer in his dissertation [25] proposed a new generalized ISD algorithm based on these two papers. In 2015, May-Ozerov [23] used the nearest neighbor algorithm to improve the BJMM version of ISD. Later in 2017, the nearest neighbor algorithm over $\mathbb{F}_q$ was applied to generalized BJMM algorithm by Gueye, Klamti and Hirose [15].

Now we focus on the first direction of improvements, which were first proposed for codes over the binary field and then later generalized over an arbitrary finite field. In 1988, the same year as Lee and Brickell proposed their algorithm, Leon [21] introduced a zero window inside the redundant set of size ℓ, where no error are allowed. In 1993 Stern [30] kept this zero window and proposed to partition the information set in to two sets and asks for v errors in each part and $t-2v$ errors outside the information set. The generalization of both Lee-Brickell and Stern's algorithm to a general finite field $\mathbb{F}_q$ were performed by Peters [27] in 2010. In 2011 Bernstein, Lange and Peters proposed the ball-collision algorithm [8], where they keep the partitioning of the information set but they reintroduce errors in the zero window, in fact they partition the zero window into two sets and ask for w errors in both and hence for $t-2v-2w$ errors outside. In 2016, Hirose [16] generalized the nearest neighbor algorithm over $\mathbb{F}_q$ and applied it to the generalized Stern algorithm.

It is important to remark (see [25]) that the BJMM algorithm, even if having the smallest complexity, comes with a different cost: memory. In order to achieve a complexity of 128 bits, BJMM needs about 10^9 terabytes of memory. In fact, Meurer observed, that if one restricts the memory to 2^{40}, BJMM and the ball-collision algorithm are performing almost the same.

In this paper we provide the missing generalization of the ball-collision algorithm. The order of the complexities of ISD algorithms over $\mathbb{F}_2$ is consistent also with their generalizations over $\mathbb{F}_q$.

3. Preliminaries

3.1. Notation

We first want to fix some notation: let q be a prime power and let $n, k, t \in \mathbb{N}$ be positive integers, such that $k, t < n$. We will denote by $\mathbf{1}_n$ the $n \times n$ identity matrix.

For an $m \times n$ matrix A and a set $S \subseteq \{1, ..., n\}$ of size k, we denote by A_S the $m \times k$ matrix consisting of the columns of A indexed by S.

For a set $S \subseteq \{1, ..., n\}$ of size k, we denote by $\mathbb{F}_q^n(S)$ the vectors in $\mathbb{F}_q^n$ having support in S. The projection of $x \in \mathbb{F}_q^n(S)$ to $\mathbb{F}_q^k$ is then canonical and denoted by $\pi_S(x)$.

On the other hand we denote by σ_S the canonical embedding of a vector $x \in \mathbb{F}_q^k$ into $\mathbb{F}_q^n(S)$, where $S \subseteq \{1, ..., n\}$ is again of size k.

For an $[n, k]$ linear code $\mathcal{C}$ over $\mathbb{F}_q$ we denote by H a parity check matrix of size $(n-k) \times n$ and by G a generator matrix of size $k \times n$. We denote the Hamming weight of a vector $x \in \mathbb{F}_q^n$, by $w(x)$. The corrupted codeword $c \in \mathbb{F}_q^n$ is given by $c = mG + e$, where $m \in \mathbb{F}_q^k$ is the message and $e \in \mathbb{F}_q^n$ is the error vector. The syndrome of c is then defined as $s = Hc^\top$ and coincides with the syndrome of the error vector, since $Hc^\top = H(mG+e)^\top = HG^\top m^\top + He^\top = He^\top$.

3.2. Ball-collision algorithm over the binary field

In what follows we describe the ball-collision algorithm over the binary proposed in [8] by Bernstein, Lange and Peters.

3.3. Concepts

There are a few concepts for computing the complexity of the ball-collision algorithm introduced in [8] that we will present and generalize to arbitrary finite fields beforehand. In general the complexity of an ISD attack consists of the cost of one iteration times the expected number of iterations. The cost in the following refers to operations, i.e. additions or multiplications, over the given field.

Algorithm 1 Ball-collision over the binary

```
Input: The $(n-k)\times n$ parity check matrix $H$, the syndrome $s \in \mathbb{F}_2^{n-k}$ and the positive integers
$v_1, v_2, w_1, w_2, k_1, k_2, \ell_1, \ell_2 \in \mathbb{Z}$, such that $k = k_1 + k_2$, $v_i \leq k_i$, $w_i \leq \ell_i$ and
$t - v_1 - v_2 - w_1 - w_2 \leq n - k - \ell_1 - \ell_2$.
Output: $e \in \mathbb{F}_2^n$ with $He^\top = s$ and $w(e) = t$.

 1: Choose $I \subseteq \{1, ..., n\}$ a uniform random information set of size $k$.
 2: Choose a uniform random partition of $I$ into disjoint sets $X_1$ and $X_2$ of size $k_1$ and $k_2 = k - k_1$ respectively.
 3: Choose uniform random partition of $Y = \{1, ..., n\} \setminus I$ into disjoint sets $Y_1, Y_2$ and $Y_3$ of sizes $\ell_1, \ell_2$ and
    $\ell_3 = n - k - \ell_1 - \ell_2$.
 4: Find an invertible matrix $U \in \mathbb{F}_2^{(n-k)\times(n-k)}$ such that $(UH)_Y = \mathbf{1}_{n-k}$ and $(UH)_I = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$, where $A_1 \in$
    $\mathbb{F}_2^{(\ell_1+\ell_2)\times k}$ and $A_2 \in \mathbb{F}_2^{\ell_3 \times k}$.
 5: Compute $Us = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$ with $s_1 \in \mathbb{F}_2^{\ell_1+\ell_2}$ and $s_2 \in \mathbb{F}_2^{\ell_3}$.
 6: Compute the set $S$ consisting of all triples $(A_1(\pi_I(x_1)) + \pi_{Y_1\cup Y_2}(y_1), x_1, y_1)$, where $x_1 \in \mathbb{F}_2^n(X_1)$, $w(x_1) = v_1$
    and $y_1 \in \mathbb{F}_2^n(Y_1)$, $w(y_1) = w_1$.
 7: Compute the set $T$ consisting of all triples $(A_1\pi_I(x_2) + \pi_{Y_1\cup Y_2}(y_2) + s_1, x_2, y_2)$, where $x_2 \in \mathbb{F}_2^n(X_2)$, $w(x_2) = v_2$
    and $y_2 \in \mathbb{F}_2^n(Y_2)$, $w(y_2) = w_2$ .
 8: for each $(a, x_1, y_1) \in S$ do
 9:     for each $(a, x_2, y_2) \in T$ do
10:         if $w(A_2(\pi_I(x_1 + x_2)) + s_2) = t - v_1 - v_2 - w_1 - w_2$: then
                Output: $e = x_1 + y_1 + x_2 + y_2 + \sigma_{Y_3}(A_2(\pi_I(x_1 + x_2)) + s_2)$.
11:         else Start over with Step 1 and a new selection of $I$.
```

i) The success probability over the binary is usually given by having chosen the correct weight distribution of the error vector. For example, if one assumes that the error vector has weight v in the information set and $t-v$ outside the information set, the success probability results in

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

This will not change over $\mathbb{F}_q$, since the scalar multiples will cross out:

$$\frac{\binom{k}{v}(q-1)^v\binom{n-k}{t-v}(q-1)^{t-v}}{\binom{n}{t}(q-1)^t} = \binom{k}{v}\binom{n-k}{t-v}\binom{n}{t}^{-1}.$$

ii) The concept of intermediate sums is important whenever one wants to compute something for all vectors in a certain space. For example we are given a $k \times n$ matrix A and want to compute $Ax^\top$ for all $x \in \mathbb{F}_2^n$, of weight t. This would usually cost k times $t-1$ additions and t 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 cost. 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 and using the sums of two columns we have already computed, means we only need to add one more column costing k additions. Proceeding in this way, until one reaches the weight t, to compute $Ax^\top$ for all $x \in \mathbb{F}_2^n$, of weight t costs $k \cdot (L(n,t) - n)$ additions, where

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

This changes slightly over a general finite field. As a first step one computes $Ax^\top$ for all $x \in \mathbb{F}_q^n$, of weight 1. Hence this step is no longer for free, but rather means computing $A\lambda$ for all $\lambda \in \mathbb{F}_q^\star$, costing $(q-1)kn$ multiplications. From there on one computes the sum of two multiples of the columns, there are $\binom{n}{2}(q-1)^2$ many and each sum costs k additions. Hence proceeding in the same manner the cost turns out to be $(q-1)kn$ multiplications and $(\bar{L}(n,t)-n(q-1))k$ additions, where

$$\bar{L}(n,t)=\sum_{i=1}^{t}\binom{n}{i}(q-1)^i.$$

iii) The next concept called early abort is also 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 $w(x+y)=t$. Hence we compute and check the weight simultaneously and if the weight of the partial solution exceeds t one does not need to continue. Over the binary 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. Over $\mathbb{F}_q$, we expect a randomly chosen element to have weight 1 with probability $\frac{q-1}{q}$, therefore we need to compute $\frac{q}{q-1}(t+1)$ many entries of the solution before we can abort.

iv) An important step in the ball-collision algorithm is to check for a collision, i.e. if $Ax^\top=By^\top$ one continues, where again $A,B\in\mathbb{F}_2^{k\times n}$ and x,y are living in some sets S and T respectively. There are $\mid S\mid\cdot\mid T\mid$ many choices for (x,y), assuming that they are distributed uniformly over $\mathbb{F}_2^n$, then on average one expects the number of collisions to be $\mid S\mid\cdot\mid T\mid 2^{-n}$. Similarly over $\mathbb{F}_q$ the number of collisions will be $\mid S\mid\cdot\mid T\mid q^{-n}$.

4. Generalization of the ball-collision algorithm

In this section we generalize the ball-collision algorithm from the binary case [8] to a general finite field.

Again, as in the binary case, the idea of the algorithm is to solve $UHe^\top=Us$ instead of $He^\top=s$, where an invertible U is chosen such that (up to permutation) $UH=\begin{pmatrix}A & \mathbf{1}_{n-k}\end{pmatrix}$ and $Us=\begin{pmatrix}s_1\\ s_2\end{pmatrix}$ with $s_1\in\mathbb{F}_q^{\ell_1+\ell_2}$, $s_2\in\mathbb{F}_q^{\ell_3}$. We are therefore looking for a vector $e\in\mathbb{F}_q^n$ fulfilling

$$UHe^\top=\begin{pmatrix}A_1 & \mathbf{1}_{\ell_1+\ell_2} & 0\\ A_2 & 0 & \mathbf{1}_{\ell_3}\end{pmatrix}\begin{pmatrix}e_1\\ e_2\\ e_3\end{pmatrix}=\begin{pmatrix}s_1\\ s_2\end{pmatrix},$$

with $e_1\in\mathbb{F}_q^k$, $e_2\in\mathbb{F}_q^{\ell_1+\ell_2}$, $e_3\in\mathbb{F}_q^{\ell_3}$. This leads to the following system of equations:

$$\begin{aligned}A_1e_1+e_2&=s_1,\\ A_2e_1+e_3&=s_2.\end{aligned}$$

The algorithm solves the above by finding

$$\begin{aligned}e_1&=\pi_I(x_1+x_2),\\ e_2&=\pi_{Y_1\cup Y_2}(y_1+y_2),\\ e_3&=-A_2(\pi_I(x_1+x_2))+s_2,\end{aligned}$$

such that

$$A_1(\pi_I(x_1))+\pi_{Y_1\cup Y_2}(y_1)=s_1-A_1(\pi_I(x_2))-\pi_{Y_1\cup Y_2}(y_2).$$

Algorithm 2 Ball-collision over $\mathbb{F}_q$

```
Input: The $(n-k)\times n$ parity check matrix $H$, the syndrome $s \in \mathbb{F}_q^{n-k}$ and the positive integers
$v_1, v_2, w_1, w_2, k_1, k_2, \ell_1, \ell_2 \in \mathbb{Z}$, such that $k = k_1 + k_2$, $v_i \leq k_i$, $w_i \leq \ell_i$ and
$t - v_1 - v_2 - w_1 - w_2 \leq n - k - \ell_1 - \ell_2$.
Output: $e \in \mathbb{F}_q^n$ with $He^\top = s$ and $w(e) = t$.

1: Choose an information set $I \subseteq \{1, ..., n\}$ of $H$ of size $k$.
2: Partition $I$ into two disjoint subsets $X_1$ and $X_2$ of size $k_1$ and $k_2 = k - k_1$ respectively.
3: Partition $Y = \{1, ..., n\} \backslash I$ into disjoint subsets $Y_1$ of size $\ell_1$, $Y_2$ of size $\ell_2$ and $Y_3$ of size $\ell_3 = n - k - \ell_1 - \ell_2$.
4: Find an invertible matrix $U \in \mathbb{F}_q^{(n-k)\times(n-k)}$, such that $(UH)_Y = \mathbf{1}_{n-k}$ and $(UH)_I = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$, where $A_1 \in \mathbb{F}_q^{(\ell_1+\ell_2)\times k}$ and $A_2 \in \mathbb{F}_q^{\ell_3 \times k}$.
5: Compute $Us = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}$, where $s_1 \in \mathbb{F}_q^{\ell_1+\ell_2}$ and $s_2 \in \mathbb{F}_q^{\ell_3}$.
6: Compute the following set:
   $$\begin{aligned} S = &\{(A_1(\pi_I(x_1)) + \pi_{Y_1 \cup Y_2}(y_1), x_1, y_1) \mid \\ &x_1 \in \mathbb{F}_q^n(X_1), w(x_1) = v_1, y_1 \in \mathbb{F}_q^n(Y_1), w(y_1) = w_1\} \end{aligned}$$
7: Compute the following set:
   $$\begin{aligned} T = &\{(-A_1(\pi_I(x_2)) + s_1 - \pi_{Y_1 \cup Y_2}(y_2), x_2, y_2) \mid \\ &x_2 \in \mathbb{F}_q^n(X_2), w(x_2) = v_2, y_2 \in \mathbb{F}_q^n(Y_2), w(y_2) = w_2\} \end{aligned}$$
8: for $(a, x_1, y_1) \in S$ do
9:     for $(a, x_2, y_2) \in T$ do
10:        if $w(-A_2(\pi_I(x_1 + x_2)) + s_2) = t - v_1 - v_2 - w_1 - w_2$ then
               Output: $e = x_1 + x_2 + y_1 + y_2 + \sigma_{Y_3}(-A_2(\pi_I(x_1 + x_2)) + s_2)$
11:        else go to Step 1 and choose new information set $I$.
```

This last condition is fulfilled by the collision between S and T in Step 9.
Observe that for $q = 2$ the above algorithm is equivalent to the one proposed over the binary. We hence did not change it in its substantial form.
We now want to prove that the ball-collision algorithm over $\mathbb{F}_q$ works, i.e. that it returns any vector e of the desired form, if it exists. For this we follow the idea of [8].

Theorem 4.1. *The ball-collision algorithm over $\mathbb{F}_q$ finds any vector e that fulfills $UHe^\top = Us$ and is of the desired form, i.e., e has v_1, v_2, w_1, w_2 and $t - v_1 - v_2 - w_1 - w_2$ nonzero entries in X_1, X_2, Y_1, Y_2 and Y_3 respectively.*

Proof. Clearly the output e is of the desired form:

- x_1 is of weight v_1 and in $\mathbb{F}_q^n(X_1)$,
- x_2 is of weight v_2 and in $\mathbb{F}_q^n(X_2)$,
- y_1 is of weight w_1 and in $\mathbb{F}_q^n(Y_1)$,
- y_2 is of weight w_2 and in $\mathbb{F}_q^n(Y_2)$,
- $w(-A_2(\pi_I(x_1 + x_2)) + s_2) = w(A_2(\pi_I(x_1 + x_2)) - s_2) = t - v_1 - v_2 - w_1 - w_2$ and it lies in $\mathbb{F}_q^n(Y_3)$.

As the above subspaces do not intersect, $w(e)$ can be calculated by adding up the weights of each of them. Hence $w(e) = t$ and each of the subspaces has the desired weight distribution by definition.

It remains to prove that $UHe^\top = Us$. Let us write each of the subspaces $\mathbb{F}_q^n(I), \mathbb{F}_q^n(Y_1 \cup Y_2)$ and $\mathbb{F}_q^n(Y_3)$ separately.

$$\begin{aligned} UHe^\top &= \begin{pmatrix} A_1 & \mathbf{1}_{\ell_1+\ell_2} & 0 \\ A_2 & 0 & \mathbf{1}_{\ell_3} \end{pmatrix} \begin{pmatrix} \pi_I(x_1+x_2) \\ \pi_{Y_1 \cup Y_2}(y_1+y_2) \\ -A_2(\pi_I(x_1+x_2)) + s_2 \end{pmatrix} \\ &= \begin{pmatrix} A_1(\pi_I(x_1+x_2)) + \pi_{Y_1\cup Y_2}(y_1+y_2) \\ A_2(\pi_I(x_1+x_2)) - A_2(\pi_I(x_1+x_2)) + s_2 \end{pmatrix} \\ &= \begin{pmatrix} A_1(\pi_I(x_1+x_2)) + \pi_{Y_1\cup Y_2}(y_1+y_2) \\ s_2 \end{pmatrix}. \end{aligned}$$

And we know that $A_1(\pi_I(x_1+x_2)) + \pi_{Y_1\cup Y_2}(y_1+y_2) = s_1$ by the collision of S and T in Step 9.

We now want to prove that the algorithm returns each of the above vectors such that $He^\top = s$ under the assumption, that we worked with a correct partitioning into X_1, X_2, Y_1, Y_2, Y_3. We do that by checking whether the algorithm considers all possible combinations and does not exclude any possible solution.

U is invertible and hence does not exclude any solution when multiplied by H and s. In Step 6, where we build the sets S and T, we go through all possible error vectors x_1, x_2, y_1 and y_2, which have the desired weight distribution. There are only two steps in the algorithm, where we exclude certain vectors:

1. When we only keep the collisions between S and T in Step 9. But this is justified as $A_1e_1 + e_2 = s_1$, i.e.

$$A_1(\pi_I(x_1)) + \pi_{Y_1\cup Y_2}(y_1) = -A_1(\pi_I(x_2)) + s_1 - \pi_{Y_1\cup Y_2}(y_2)$$

   needs to be satisfied.

2. When we check whether $w(-A_2(\pi_I(x_1+x_2)) + s_2) = t - v_1 - v_2 - w_1 - w_2$. But also this is justified as $e_3 \in \mathbb{F}_q^{\ell_3}$ needs this weight to complete the weight of e to be t.

Hence we consider all possible error vectors that are of the given weight distribution and satisfy $UHe^\top = Us$. □

5. Complexity analysis

In this section we want to analyze the complexity of the extended ball-collision algorithm over $\mathbb{F}_q$. Since the cost will be given in operations over $\mathbb{F}_q$, we will denote by $\mathcal{M}$ the multiplications needed and by $\mathcal{A}$ the amount of additions. Note that one addition over $\mathbb{F}_q$ costs $\log_2(q)$ bit operations and one multiplication over $\mathbb{F}_q$ costs $\log_2(q)^2$ bit operations, observe that one could also use the following speed up [29] such that multiplication costs $\log_2(q)\log_2(\log_2(q))\log_2(\log_2(\log_2(q)))$ bit operations.

Success Probability of one Iteration

We have the same success probability over $\mathbb{F}_q$ as over $\mathbb{F}_2$, as observed in Section 3.3, hence one iteration succeeds with a probability of

$$\binom{n}{t}^{-1}\binom{\ell_3}{t-v_1-v_2-w_1-w_2}\binom{k_1}{v_1}\binom{k_2}{v_2}\binom{\ell_1}{w_1}\binom{\ell_2}{w_2}. \tag{1}$$

Cost of one Iteration

1. In Step 4 of the algorithm, one uses Gaussian elimination to find an invertible matrix U, bringing H into systematic form, since we will also need to compute Us we will directly perform Gaussian elimination on the matrix $\begin{pmatrix} H \mid s \end{pmatrix}$, where we adjoined the vector s as a column to H. A broad estimate of the cost for this step is $(n-k)^2(n+1)(\mathcal{A}+\mathcal{M})$.

2. To build the set S we want to use the concept of intermediate sums over $\mathbb{F}_q$ described before. Hence to compute $A_1(\pi_I(x_1))$, for all $x_1 \in \mathbb{F}_q^n(X_1)$ we need $(q-1)(\ell_1+\ell_2)k_1$ multiplications and $(\bar{L}(k_1,v_1)-k_1(q-1))(\ell_1+\ell_2)$ additions. After having computed $A_1(\pi_I(x_1))$, we add $\pi_{Y_1\cup Y_2}(y_1)$ using intermediate sums. This costs $\bar{L}(\ell_1,w_1)$ additions for each of the $x_1 \in \mathbb{F}_q^n(X_1)$, which are $\binom{k_1}{v_1}(q-1)^{v_1}$ many. Hence resulting in a total cost of
$$\begin{aligned}&(q-1)(\ell_1+\ell_2)k_1\mathcal{M} + (\bar{L}(k_1,v_1)-k_1(q-1))(\ell_1+\ell_2)\mathcal{A}\\ &+\binom{k_1}{v_1}\bar{L}(\ell_1,w_1)(q-1)^{v_1}\mathcal{A}.\end{aligned}$$

3. To build the set T we proceed similarly, the only difference being that s_1 needs to be added to the first step of the intermediate sums over $\mathbb{F}_q$, hence adding a cost of $(\ell_1+\ell_2)(q-1)k_2$ additions. The total cost of this step is hence given by
$$\begin{aligned}&(q-1)(\ell_1+\ell_2)k_2(\mathcal{M}+\mathcal{A}) + (\bar{L}(k_2,v_2-k_2(q-1)))(\ell_1+\ell_2)\mathcal{A}\\ &+\binom{k_2}{v_2}\bar{L}(\ell_2,w_2)(q-1)^{v_2}\mathcal{A}.\end{aligned}$$

4. In Step 9, when checking for collisions between S and T, we want to calculate the number of collisions we can expect on average. The elements in S and T are all of length $\ell_1+\ell_2$ and hence there is a total of $q^{\ell_1+\ell_2}$ possible elements. S has $\binom{k_1}{v_1}\binom{\ell_1}{w_1}(q-1)^{v_1+w_1}$ many elements and T has $\binom{k_2}{v_2}\binom{\ell_2}{w_2}(q-1)^{v_2+w_2}$ many elements, we therefore get that the expected number of collisions is
$$\frac{\binom{k_1}{v_1}\binom{k_2}{v_2}\binom{\ell_1}{w_1}\binom{\ell_2}{w_2}(q-1)^{v_1+v_2+w_1+w_2}}{q^{\ell_1+\ell_2}}.$$

5. For each collision we have, we check whether $w(-A_2(\pi_I(x_1+x_2))+s_2) = t-v_1-v_2-w_1-w_2$ is satisfied. For this we will use the method of early abort: to compute one bit of the result costs (v_1+v_2+1) additions and (v_1+v_2) multiplications, hence this step costs on average
$$\frac{q}{q-1}(t-v_1-v_2-w_1-w_2+1)\left((v_1+v_2+1)\mathcal{A}+(v_1+v_2)\mathcal{M}\right).$$

Hence the total cost of one iteration is given by

$$\begin{aligned}&(n-k)^2(n+1)(\mathcal{A}+\mathcal{M})\\ &+(\ell_1+\ell_2)[(q-1)((k_1+k_2)\mathcal{M}+k_2\mathcal{A})\\ &+(\bar{L}(k_1,v_1)-k_1(q-1))\mathcal{A}+(\bar{L}(k_2,v_2)-k_2(q-1))\mathcal{A}]\\ &+\binom{k_1}{v_1}\bar{L}(\ell_1,w_1)(q-1)^{v_1}\mathcal{A}+\binom{k_2}{v_2}\bar{L}(\ell_2,w_2)(q-1)^{v_2}\mathcal{A}\\ &+\binom{k_1}{v_1}\binom{k_2}{v_2}\binom{\ell_1}{w_1}\binom{\ell_2}{w_2}(q-1)^{v_1+v_2+w_1+w_2}q^{-(\ell_1+\ell_2)}\\ &\cdot\frac{q}{q-1}(t-v_1-v_2-w_1-w_2+1)\left((v_1+v_2+1)\mathcal{A}+(v_1+v_2)\mathcal{M}\right).\end{aligned} \tag{2}$$

Overall cost

Combining the result from (1) and (2) the overall cost of the ball-collision algorithm over $\mathbb{F}_q$ then amounts to

$$\begin{aligned}
&\binom{n}{t}\left(\binom{\ell_3}{t-v_1-v_2-w_1-w_2}\binom{k_1}{v_1}\binom{k_2}{v_2}\binom{\ell_1}{w_1}\binom{\ell_2}{w_2}\right)^{-1}\\
&\cdot [(n-k)^2(n+1)(\mathcal{A}+\mathcal{M})\\
&+(\ell_1+\ell_2)[(q-1)((k_1+k_2)\mathcal{M}+k_2\mathcal{A})\\
&+(\bar{L}(k_1,v_1)-k_1(q-1))\mathcal{A}+(\bar{L}(k_2,v_2)-k_2(q-1))\mathcal{A}]\\
&+\binom{k_1}{v_1}\bar{L}(\ell_1,w_1)(q-1)^{v_1}\mathcal{A}+\binom{k_2}{v_2}\bar{L}(\ell_2,w_2)(q-1)^{v_2}\mathcal{A}\\
&+\binom{k_1}{v_1}\binom{k_2}{v_2}\binom{\ell_1}{w_1}\binom{\ell_2}{w_2}(q-1)^{v_1+v_2+w_1+w_2}q^{-(\ell_1+\ell_2)}\\
&\cdot\frac{q}{q-1}(t-v_1-v_2-w_1-w_2+1)\left((v_1+v_2+1)\mathcal{A}+(v_1+v_2)\mathcal{M}\right)].
\end{aligned}$$

5.1. Asymptotic complexity

In this subsection we want to find the asymptotic complexity of the ball-collision algorithm over $\mathbb{F}_q$.

Fix real numbers $0 < T < 1/2$ and R, with

$$-T\log_q(T)-(1-T)\log_q(1-T)\leq 1-R<1.$$

We consider codes of large length n, we fix functions $k,t:\mathbb{N}\to\mathbb{N}$ which satisfy $\lim_{n\to\infty} t(n)/n=T$ and $\lim_{n\to\infty} k(n)/n=R$.

We fix real numbers V,W,L with $0\leq V\leq R/2, 0\leq W\leq L$ and

$$0\leq T-2V-2W\leq 1-R-2L.$$

We fix the parameters $v_1,v_2,w_1,w_2,\ell_1,\ell_2,k_1,k_2$ of the ball-collision algorithm over $\mathbb{F}_q$ such that

i) $\lim_{n\to\infty}\frac{v_i}{n}=V$,

ii) $\lim_{n\to\infty}\frac{w_i}{n}=W$,

iii) $\lim_{n\to\infty}\frac{k_i}{n}=R/2$,

iv) $\lim_{n\to\infty}\frac{\ell_i}{n}=L$,

for $i\in\{1,2\}$. We use the convention that $x\log_q(x)=0$, for $x=0$. In what follows we will use the following asymptotic formula for binomial coefficients:

$$\lim_{n\to\infty}\frac{1}{n}\log_q\binom{\alpha+o(1)n}{\beta+o(1)n}=\alpha\log_q(\alpha)-\beta\log_q(\beta)-(\alpha-\beta)\log_q(\alpha-\beta).$$

With this formula we get the following:

i) $\lim_{n\to\infty}\frac{1}{n}\log_q\binom{n}{t}=-T\log_q(T)-(1-T)\log_q(1-T)$,

ii) $\lim_{n\to\infty}\frac{1}{n}\log_q\binom{k_i}{v_i}=R/2\log_q(R/2)-V\log_q(V)-(R/2-V)\log_q(R/2-V)$,

iii) $\lim_{n\to\infty} \frac{1}{n} \log_q \binom{\ell_i}{w_i} = L\log_q(L) - W\log_q(W) - (L-W)\log_q(L-W)$,

iv) $\lim_{n\to\infty} \frac{1}{n} \log_q \binom{n-k-\ell_1-\ell_2}{t-v_1-v_2-w_1-w_2} = (1-R-2L)\log_q(1-R-2L) - (T-2V-2W)\log_q(T-2V-2W) - (1-R-2L-T+2V+2W)\log_q(1-R-2L-T+2V+2W)$.

Success probability

We will denote by $S(V,W,L)$ the asymptotic exponent of the success probability:

$$\begin{aligned}
S(V,W,L) &= \lim_{n\to\infty} \frac{1}{n} \log_q \left(\binom{n}{t}^{-1} \binom{n-k-\ell_1-\ell_2}{t-v_1-v_2-w_1-w_2} \binom{k_1}{v_1}\binom{k_2}{v_2}\binom{\ell_1}{w_1}\binom{\ell_2}{w_2} \right) \\
&= T\log_q(T) + (1-T)\log_q(1-T) + (1-R-2L)\log_q(1-R-2L) \\
&\quad -(T-2V-2W)\log_q(T-2V-2W) \\
&\quad -(1-R-2L-T+2V+2W)\log_q(1-R-2L-T+2V+2W) \\
&\quad +R\log_q(R/2) - 2V\log_q(V) - (R-2V)\log_q(R/2-V) + 2L\log_q(L) \\
&\quad -2W\log_q(W) - 2(L-W)\log_q(L-W).
\end{aligned}$$

Cost of one iteration

We will denote by $C(V,W,L)$ the asymptotic exponent of the cost of one iteration.

$$\begin{aligned}
C(V,W,L) &= \lim_{n\to\infty} \frac{1}{n} \log_q \left(\binom{k_1}{v_1}(q-1)^{v_1} + \binom{k_2}{v_2}(q-1)^{v_2} + \binom{k_1}{v_1}\binom{\ell_1}{w_1}(q-1)^{v_1+w_1} \right. \\
&\quad \left. + \binom{k_2}{v_2}\binom{\ell_2}{w_2}(q-1)^{v_2+w_2} + \binom{k_1}{v_1}\binom{k_2}{v_2}\binom{\ell_1}{w_1}\binom{\ell_2}{w_2}(q-1)^{v_1+v_2+w_1+w_2}q^{-\ell_1-\ell_2} \right) \\
&= \max\left\{\log_q(q-1)V + R/2\log_q(R/2) - V\log_q(V) - (R/2-V)\log_q(R/2-V), \right. \\
&\quad \log_q(q-1)(V+W) + R/2\log_q(R/2) - V\log_q(V) - (R/2-V)\log_q(R/2-V) \\
&\quad +L\log_q(L) - W\log_q(W) - (L-W)\log_q(L-W), \\
&\quad \log_q(q-1)(2V+2W) - 2L + R\log_q(R/2) - 2V\log_q(V) \\
&\quad \left. -(R-2V)\log_q(R/2-V) + 2L\log_q(L) - 2W\log_q(W) - (2L-2W)\log_q(L-W)\right\}.
\end{aligned}$$

Overall cost

The overall asymptotic cost exponent of the ball-collision algorithm over $\mathbb{F}_q$ is given by the difference of $C(V,W,L)$ and $S(V,W,L)$:

$$\begin{aligned}
D(V,W,L) &= \max\left\{\log_q(q-1)V - R/2\log_q(R/2) + V\log_q(V) + (R/2-V)\log_q(R/2-V) \right. \\
&\quad -2L\log_q(L) + 2W\log_q(W) + 2(L-W)\log_q(L-W), \\
&\quad \log_q(q-1)(V+W) - R/2\log_q(R/2) + V\log_q(V) + (R/2-V)\log_q(R/2-V) \\
&\quad -L\log_q(L) + W\log_q(W) + (L-W)\log_q(L-W), \\
&\quad \left. \log_q(q-1)(2V+2W) - 2L\right\} - T\log_q(T) - (1-T)\log_q(1-T) \\
&\quad (1-R-2L)\log_q(1-R-2L) + (T-2V-2W)\log_q(T-2V-2W) \\
&\quad +(1-R-2L-T+2V+2W)\log_q(1-R-2L-T+2V+2W).
\end{aligned}$$

The asymptotic complexity is then given by $q^{D(V,W,L)n+o(n)}$.

Asymptotically, we assume that the code attains the Gilbert-Varshamov bound, i.e. the code rate

$R = k/n$ and the distance $D = d/n$ relate via:

$$R = 1 + D\log_q(D) + (1-D)\log_q(1-D) - D\log_q(q-1). \tag{3}$$

In order to compute the asymptotic complexity of half-distance decoding (i.e. $T = D/2$) for a fixed rate R, we performed a numerical optimization of the parameters V, W and L such that the overall cost $D(V, W, L)$ is minimized subject to the following constraints:

$$0 \leq V \leq R/2, 0 \leq W \leq L \text{ and } 0 \leq T - 2V - 2W \leq 1 - R - 2L.$$

Let $F(q, R)$ be the exponent of the optimized asymptotic complexity. The asymptotic complexity of half-distance decoding at rate R over $\mathbb{F}_q$ is then given by $q^{F(q,R)n+o(n)}$.

In Table 1, the values refer to the exponent of the worst-case complexity of distinct algorithms, i.e. $F(q, R_w)$ where $R_w = \text{argmax}_{0<R<1}(F(q, R))$. It compares Peter's generalization of Stern's algorithm to $\mathbb{F}_q$, Hirose 's generalization of Stern's algorithm using May-Ozerov's nearest neighbor algorithm (MO) to $\mathbb{F}_q$, Gueye *et al.* generalization of the algorithm of BJMM using MO to $\mathbb{F}_q$ and the generalization of the ball-collision algorithm to $\mathbb{F}_q$.

Table 1: Comparison of the asymptotic complexities over $\mathbb{F}_q$. The values $q-$Stern and $q-$Stern-MO are from [16, Table 1], and $q-$BJMM-MO are from [15, Table3].

q	$q-$Stern	$q-$Stern-MO	$q-$BJMM-MO	$q-$Ball-collision
2	0.05563	0.05498	0.04730	0.055573
3	0.05217	0.05242	0.04427	0.052145
4	0.04987	0.05032	0.04294	0.049846
5	0.04815	0.04864	0.03955	0.048140
7	0.04571	0.04614	0.03706	0.045697
8	0.04478	0.04519	0.03593	0.044770
11	0.04266	0.04299	0.03335	0.042656

We can observe that the ball-collision algorithm over $\mathbb{F}_q$ outperforms Peter's generalization of Stern's algorithm to $\mathbb{F}_q$ and Hirose's ISD algorithm over $\mathbb{F}_q$, for all $q \geq 3$. Like in the binary case, the ball-collision algorithm does not outperform the generalization of Gueye *et al.* of the BJMM algorithm using MO to $\mathbb{F}_q$.

Acknowledgment: The fourth author is thankful to the Swiss National Science Foundation grant number 169510.

References

[1] A. Al Jabri, A statistical decoding algorithm for general linear block codes, In IMA International Conference on Cryptography and Coding, pages 1–8, Springer, 2001.

[2] A. Ashikhmin, A. Barg, Minimal vectors in linear codes, IEEE Trans. Inform. Theory 44(5) (1998) 2010–2017.

[3] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, D. Schipani, Enhanced public key security for the McEliece cryptosystem, J. Cryptology 29 (2016) 1–27.

[4] G. Banegas, P. SLM Barreto, B. Odilon Boidje, P.-L. Cayrel, G. Ndollane Dione, K. Gaj, C. Thiécoumba Gueye, R. Haeussler, J. Belo Klamti, O. N'diaye, et al. DAGS: Key encapsulation using dyadic GS codes, J. Math. Cryptol. 12(4) (2018) 221–239.

[5] A. Barg, E. Krouk, H. C. A. van Tilborg, On the complexity of minimum distance decoding of long linear codes, IEEE Trans. Inform. Theory 45(5) (1999) 1392–1405.

[6] A. Becker, A. Joux, A. May, A. Meurer, Decoding random binary linear codes in 2n/20: How 1+ 1= 0 improves information set decoding, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 520–536. Springer, 2012.

[7] E. Berlekamp, R. McEliece, H. van Tilborg, On the inherent intractability of certain coding problems (Corresp.), IEEE Trans. Inform. Theory 24(3) (1978) 384–386.

[8] D. J. Bernstein, T. Lange, C. Peters, Smaller decoding exponents: ball-collision decoding, In Annual Cryptology Conference, pages 743–760, Springer, 2011.

[9] J. Bolkema, H. Gluesing-Luerssen, C. A. Kelley, K. Lauter, B. Malmskog, J. Rosenthal. Variations of the McEliece Cryptosystem, In Algebraic Geometry for Coding Theory and Cryptography, pages 129–150. Springer, 2017.

[10] A. Canteaut, 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 Trans. Inform. Theory 44(1) (1998) 367–378.

[11] A. Canteaut, N. Sendrier, Cryptanalysis of the original McEliece cryptosystem, In International Conference on the Theory and Application of Cryptology and Information Security, pages 187–199. Springer, 1998.

[12] F. Chabaud, Asymptotic analysis of probabilistic algorithms for finding short codewords, In Eurocode'92, pages 175–183, Springer, 1993.

[13] I. I. Dumer, Two decoding algorithms for linear codes, Problemy Peredachi Informatsii, 25(1) (1989) 24–32.

[14] M. Finiasz, N. Sendrier, Security bounds for the design of code-based cryptosystems, In International Conference on the Theory and Application of Cryptology and Information Security, pages 88–105, Springer, 2009.

[15] C. T. Gueye, J. B. Klamti, S. Hirose, Generalization of BJMM-ISD using May-Ozerov nearest neighbor algorithm over an arbitrary finite field $\mathbb{F}_q$, In Said El Hajji, Abderrahmane Nitaj, and El Mamoun Souidi, editors, Codes, Cryptology and Information Security, pages 96–109, Springer, 2017.

[16] 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, pages 115–126, Springer, 2016.

[17] N. Howgrave-Graham, A. Joux, New generic algorithms for hard knapsacks, In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 235–256. Springer, 2010.

[18] K. Khathuria, J. Rosenthal, V. Weger, Weight Two Masking of the Reed- Solomon Structure in Conjugation with List Decoding. In Proceedings of 23rd International Symposium on Mathematical Theory of Networks and Systems, pages 309–314, Hong Kong University of Science and Technology, Hong Kong, 2018.

[19] E. A. Kruk, Decoding complexity bound for linear block codes, Probl. Peredachi Inf. 25(3) (1989) 103–107.

[20] P. J. Lee, 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, pages 275–280, Springer, 1988.

[21] J. S. Leon, A probabilistic algorithm for computing minimum weights of large error–correcting codes, IEEE Trans. Inform. Theory 34(5) (1988) 1354–1359.

[22] A. May, A. Meurer, 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, pages 107–124, Springer, 2011.

[23] A. May, 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, pages 203–228. Springer, 2015.

[24] R. J. McEliece, A Public-Key Cryptosystem Based on Algebraic Coding Theory, Technical report,

DSN Progress report, Jet Propulsion Laboratory, Pasadena, 1978.

[25] A. Meurer, A coding-theoretic approach to cryptanalysis, PhD thesis, Bochum-Ruhr University, 2012.

[26] R. Niebuhr, E. Persichetti, P.-L. Cayrel, S. Bulygin, J. Buchmann, On lower bounds for information set decoding over $\mathbb{F}_q$ and on the effect of partial knowledge, Int. J. Inf. Coding Theory 4(1) (2017) 47–78.

[27] C. Peters, Information-set decoding for linear codes over $\mathbb{F}_q$, In International Workshop on Post-Quantum Cryptography, pages 81–94, Springer, 2010.

[28] E. Prange, The use of information sets in decoding cyclic codes, IRE Transactions on Information Theory 8(5) (1962) 5–9.

[29] A. Schönhage, V. Strassen, Schnelle multiplikation großer zahlen, Computing 7(3) (1971) 281—292.

[30] J. Stern, A new identification scheme based on syndrome decoding, In Annual International Cryptology Conference, pages 13–21, Springer, 1993.

[31] J. van Tilburg, On the McEliece public-key cryptosystem, In Conference on the Theory and Application of Cryptography, pages 119–131, Springer, 1988.