May-Ozerov Algorithm for Nearest-Neighbor Problem over $\mathbb{F}_q$ and Its Application to Information Set Decoding

Shoichi Hirose

Graduate School of Engineering, University of Fukui, Japan

hrs_shch@u-fukui.ac.jp

Abstract. May and Ozerov proposed an algorithm for the nearest-neighbor problem of vectors over the binary field at EUROCRYPT 2015. They applied their algorithm to the decoding problem of random linear codes over the binary field and confirmed the performance improvement. We describe their algorithm generalized to work for vectors over the finite field $\mathbb{F}_q$ with arbitrary prime power q. We also apply the generalized algorithm to the decoding problem of random linear codes over $\mathbb{F}_q$. It is observed by our numerical analysis of asymptotic time complexity that the May-Ozerov nearest-neighbor algorithm may not contribute to the performance improvement of the Stern information set decoding over $\mathbb{F}_q$ with $q \geq 3$.

Keywords: code-based cryptography, random linear code, information set decoding, nearest-neighbor problem

1 Introduction

Background. Decoding random linear codes is a well-known combinatorial problem in coding theory and cryptography. No efficient algorithm is found for this problem, and the intractability is used to construct various cryptographic schemes. In particular, different from public key cryptosystems based on factoring or discrete logarithms, public key cryptosystems based on codes such as McEliece PKC [10] are expected to remain secure even if large-scale quantum computers become available.

An $[n, k]$ linear code over the finite field $\mathbb{F}_q$ is a k-dimensional subspace of $\mathbb{F}_q^n$. n is the length of the code and k/n is called the rate. An $[n, k]$ linear code over $\mathbb{F}_q$ can be defined as a kernel of a matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ with rank $n - k$. $\boldsymbol{H}$ is called a parity check matrix. The distance d of an $[n, k]$ linear code is the minimum Hamming distance between its codewords.

A random parity check matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ specifies a random $[n, k]$ linear code. It is shown that, for large n, virtually all random linear $[n, k]$ codes over $\mathbb{F}_q$ achieve the Gilbert-Varshamov bound $k/n \leq 1 - H_q(d/n)$, where H_q is the q-ary entropy function [3]. Thus, it is assumed in this paper that d satisfies $k/n = 1 - H_q(d/n)$.

An instance of the decoding problem of random linear codes is a pair of a random parity check matrix $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$ and a vector $\boldsymbol{x} \in \mathbb{F}_q^n$. The required answer is a codeword with minimum Hamming distance to $\boldsymbol{x}$. This setting is called the full distance decoding. The other setting, which is more typical in the application to cryptography, promises that there exists a codeword $\boldsymbol{c}$ such that $\boldsymbol{x} = \boldsymbol{c}+\boldsymbol{e}$ and the Hamming weight of $\boldsymbol{e}$ is less than or equal to $\lfloor (d-1)/2 \rfloor$, where d is the distance of the given code. This setting is called the bounded distance decoding, which will be focused on in this paper. In the bounded distance decoding, it is ensured that the answer $\boldsymbol{c}$ is unique.

Related Work. The important class of algorithms for decoding random linear codes is information set decoding (ISD), which was first suggested by Prange [13]. ISD consists of two steps: the first step is a permutation step and the second step is a search step. A successive execution of these steps is iterated until an answer is obtained. In its basic form by Lee and Brickell [7], in the first permutation step, one first permutes the columns of $\boldsymbol{H}$ randomly and transform

the permuted $\boldsymbol{H}$ into $(\boldsymbol{R}\ \boldsymbol{I})$ with Gaussian elimination, where $\boldsymbol{R} \in \mathbb{F}_q^{(n-k)\times k}$ and $\boldsymbol{I}$ is the $(n-k)$-dimensional identity matrix. The Gaussian elimination is also applied to the syndrome $\boldsymbol{s} = \boldsymbol{H}\boldsymbol{x}$, which is transformed to $\tilde{\boldsymbol{s}}$. In the second search step, for some fixed p, one searches a linear combination of p columns of $\boldsymbol{R}$ whose Hamming distance to $\tilde{\boldsymbol{s}}$ is $w-p$. For such a linear combination, $\tilde{\boldsymbol{s}}$ is obtained by adding a linear combination of $w-p$ columns of $\boldsymbol{I}$ to the linear combination. Thus, one can recover $\boldsymbol{e}$ and obtain $\boldsymbol{c} = \boldsymbol{x} - \boldsymbol{e}$. p is chosen to optimize the time complexity.

Stern reduced the time complexity of ISD using the meet-in-the-middle approach for the search step [14]. The Stern ISD is the best algorithm in terms of time complexity for about twenty years. Recently, several proposals for the search step have been made to further reduce the time complexity. Bernstein, Lange and Peters introduced the ball-collision technique [2]. May, Meurer and Thomae [8] used the representation technique introduced by Howgrave-Graham and Joux [6] for the subset sum problem. Becker, Joux, May and Meurer [1] introduced an interesting tweak to the algorithm by May, et al. [8]. May-Ozerov devised an algorithm to find a pair of nearest neighbors [9].

The decoding problem of random linear codes is often discussed for codes over the binary field. Still, some work has been done to generalize ISD for codes over other finite fields. Coffey and Goodman [4] analyzed the complexity of the Prange ISD over $\mathbb{F}_q$. Peters [12] generalized the Stern ISD and its extension by Finiansz and Sendrier [5]. Meurer [11] generalized the BJMM ISD [1] and analyzed its time complexity. May and Ozerov [9] claimed that they did not see any obstacles in transferring their algorithm to $\mathbb{F}_q$. However, the generalization does not seem so straightforward as the generalization of the other algorithms.

Our Contribution. In this paper, the May-Ozerov algorithm for the nearest-neighbor problem is generalized to work over $\mathbb{F}_q$ with any prime power q. The time complexity of the algorithm is also analyzed. The analysis suggests that the May-Ozerov algorithm may not be practical even for small $q \geq 3$ due to the factors of the time complexity which does not appear in its $\tilde{O}$-notation. Then, the May-Ozerov algorithm is applied to the decoding problem of random linear codes over $\mathbb{F}_q$. The asymptotic time complexity of the Stern ISD with the May-Ozerov nearest-neighbor algorithm is analyzed by numerical optimization. It is observed by the analysis that the May-Ozerov nearest-neighbor algorithm may not contribute to the performance improvement of the Stern ISD over $\mathbb{F}_q$ with $q \geq 3$.

Organization. The paper is organized as follows. Section 2 gives some notations and definitions. The May-Ozerov algorithm for the nearest-neighbor problem over $\mathbb{F}_q$ is presented in Sect. 3. The application of the May-Ozerov algorithm to the Stern ISD over $\mathbb{F}_q$ is described in Sect. 4. Some numerical analyses of asymptotic time complexity of this algorithm is given in Sect. 5. A concluding remark is given in Sect. 6.

2 Preliminaries

2.1 Notation

The q-ary entropy function is denoted by H_q. Namely,

$$H_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x)\ .$$

The binary entropy function H_2 is often simply denoted by H.

Let $\mathbb{F}_q$ be the finite field for prime power q. $\mathbb{F}_q$ is also used to represent the set of elements of the field.

Let $\boldsymbol{w} \in \mathbb{F}_q^l$ be a vector. The Hamming weight of $\boldsymbol{w}$ is the number of its nonzero coordinates, which is denoted by $w_{\mathrm{H}}(\boldsymbol{w})$. The number of the coordinates of $\boldsymbol{w}$ is denoted by $|\boldsymbol{w}|$, that is, $|\boldsymbol{w}| = l$.

2.2 Multinomial Coefficient and Stirling's Formula

The multinomial coefficient

$$\binom{n}{n_1, n_2, \cdots, n_\tau} = \frac{n!}{n_1! n_2! \cdots n_\tau!}$$

is the number of ways to split n distinct elements into τ disjoint groups with the size of the i-th group n_i for $1 \le i \le \tau$, where $n = n_1 + n_2 + \cdots + n_\tau$ and $\tau \ge 2$.

We will often use Stirling's formula $n! = \sqrt{2\pi n}(n/\mathrm{e})^n \mathrm{e}^{o(1)}$ and

$$\binom{\kappa n}{\mu n} = \sqrt{\frac{\kappa}{2\pi\mu(\kappa-\mu)}} 2^{\kappa H(\mu/\kappa)n - o(n)} = \tilde{\Theta}\left(2^{\kappa H(\mu/\kappa)n}\right) \ .$$

2.3 Nearest-Neighbor Problem over $\mathbb{F}_q$

The nearest-neighbor (NN) problem over the binary field defined in [9] is generalized over other finite fields:

Definition 1 (Nearest-Neighbor Problem over $\mathbb{F}_q$). *Let q be a prime power. Let m be a positive integer. Let $0 < \gamma < 1/2$ and $0 < \lambda < 1$. The (m, γ, λ)-NN problem over $\mathbb{F}_q$ is defined as follows:*

Input *$\mathcal{U}$, $\mathcal{V}$ and γ, where $\mathcal{U} \subset \mathbb{F}_q^m$, $\mathcal{V} \subset \mathbb{F}_q^m$ and $|\mathcal{U}| = |\mathcal{V}| = q^{\lambda m}$,*
Output *$\mathcal{C} \subset \mathcal{U} \times \mathcal{V}$ which have $(\boldsymbol{u}^*, \boldsymbol{v}^*)$ such that $w_{\mathrm{H}}(\boldsymbol{u}^* - \boldsymbol{v}^*) = \gamma m$ (if any).*

It is also assumed that the vectors in $\mathcal{U}$ and $\mathcal{V}$ are chosen uniformly at random and pairwise independent.

To simplify the description of the May-Ozerov algorithm for the NN problem, the balancedness of a vector over $\mathbb{F}_q$ is defined:

Definition 2. *A vector in $\mathbb{F}_q^l$ is called balanced if the number of its coordinates equal to x is l/q for every element $x \in \mathbb{F}_q$.*

3 May-Ozerov Algorithm for Nearest-Neighbor Problem over $\mathbb{F}_q$

The May-Ozerov algorithm for the nearest-neighbor problem over $\mathbb{F}_2$ [9] is generalized to work over $\mathbb{F}_q$ with arbitrary prime power q. The generalized algorithm is given in Algorithm 1. An overview of the algorithm is given below.

For a given pair of lists, $\mathcal{U}$ and $\mathcal{V}$, the May-Ozerov NN algorithm creates exponentially many pairs of sublists with sizes expected polynomial so that at least one of the pairs of sublists contain an unknown solution with overwhelming probability. Since the sizes of the sublists are expected to be polynomial, the naive search is carried out to find the unknown solution.

All the vectors in the given lists first randomized with a random permutation matrix $\boldsymbol{P}$ and a random vector $\boldsymbol{r}$. This randomization plays an important role in the algorithm. In the description of Algorithm 1,

$$\boldsymbol{P}\mathcal{U} + \boldsymbol{r} = \{\boldsymbol{u}' \,|\, \boldsymbol{u}' = \boldsymbol{P}\boldsymbol{u} + \boldsymbol{r},\ \boldsymbol{u} \in \mathcal{U}\} \ ,$$

and $\boldsymbol{P}\mathcal{V} + \boldsymbol{r}$ is defined similarly. $\boldsymbol{P}$ is used for random transposition of coordinates of each vector.

Each pair of sublists are created first by choosing some of the coordinates of the vectors at random. Let A be the set of the chosen coordinates with $|A| = \beta m$ for $0 < \beta < 1$. Then, a pair of sublists consist of vectors satisfying that the number of coordinates in A equal to $x \in \mathbb{F}_q$ is $h_x \beta m$, where h_x's are positive and $\sum_{x \in \mathbb{F}_q} h_x = 1$. Actually, the vectors are filtered gradually

with recursive calls to the procedure NNR. Each vector is divided into t pieces with the size of the i-th piece $\alpha_i m$, where α_i's are positive and $\alpha_1 + \alpha_2 + \cdots + \alpha_t = 1$. A is a union of disjoint sets $A_1, A_2, \ldots, A_t$, where the coordinates in A_i are from the i-th piece and $|A_i| = \beta\alpha_i m$ for $1 \leq i \leq t$. The pair of sublists consist of vectors satisfying that the number of coordinates in A_i equal to $x \in \mathbb{F}_q$ is $h_x\beta\alpha_i m$ for $1 \leq i \leq t$.

The recursive calls to the procedure NNR form a tree structure with the root corresponding to the call from MO-NN. The last argument i of NNR represents the depth of the call in the tree, where the depth of the root is 1 and the depth of a leaf is $t + 1$.

Algorithm 1 May-Ozerov Algorithm for (m, γ, λ)-NN problem over $\mathbb{F}_q$

procedure MO-NN($\mathcal{U}, \mathcal{V}, \gamma$)
  $y \leftarrow (1-\gamma)\left(H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q\left(\frac{qh_x-\gamma}{1-\gamma}\beta\right)\right)$
  Select $\varepsilon > 0$
  $t \leftarrow \lceil(\log_2(y - \lambda + \varepsilon/2) - \log_2(\varepsilon/2))/(\log_2 y - \log_2 \lambda)\rceil$
  $\alpha_1 \leftarrow (y - \lambda + \varepsilon/2)/y$
  $\alpha_i \leftarrow (\lambda/y)\alpha_{i-1}$ for $2 \leq i \leq t$
  for $m^{O(1)}$ times **do**
    Select a permutation matrix $\boldsymbol{P} \in \{0,1\}^{m\times m}$ u.a.r. ▷ u.a.r. means uniformly at random.
    Select u.a.r. $\boldsymbol{r} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_t) \in \mathbb{F}_q^m$ s.t. $\boldsymbol{r}_i \in \mathbb{F}_q^{\alpha_i m}$ is balanced for every $1 \leq i \leq t$
    $\tilde{\mathcal{U}} \leftarrow \{\tilde{\boldsymbol{u}} \,|\, \tilde{\boldsymbol{u}} \in \boldsymbol{P}\mathcal{U} + \boldsymbol{r} \wedge \forall j.(\tilde{\boldsymbol{u}}_j \in \mathbb{F}_q^{\alpha_j m} \text{ is balanced})\}$ ▷ $\tilde{\boldsymbol{u}} = (\tilde{\boldsymbol{u}}_1, \ldots, \tilde{\boldsymbol{u}}_t)$ and $1 \leq j \leq t$
    $\tilde{\mathcal{V}} \leftarrow \{\tilde{\boldsymbol{v}} \,|\, \tilde{\boldsymbol{v}} \in \boldsymbol{P}\mathcal{V} + \boldsymbol{r} \wedge \forall j.(\tilde{\boldsymbol{v}}_j \in \mathbb{F}_q^{\alpha_j m} \text{ is balanced})\}$ ▷ $\tilde{\boldsymbol{v}} = (\tilde{\boldsymbol{v}}_1, \ldots, \tilde{\boldsymbol{v}}_t)$ and $1 \leq j \leq t$
    return NNR($\tilde{\mathcal{U}}, \tilde{\mathcal{V}}, m, t, \gamma, \lambda, \alpha_1, \ldots, \alpha_t, y, \varepsilon, 1$)
  end for
end procedure

procedure NNR($\tilde{\mathcal{U}}, \tilde{\mathcal{V}}, m, t, \gamma, \lambda, \alpha_1, \ldots, \alpha_t, y, \varepsilon, i$))
  if $i = t + 1$ **then**
    $\mathcal{C} \leftarrow \{(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) \,|\, (\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) \in \tilde{\mathcal{U}} \times \tilde{\mathcal{V}} \wedge w_{\mathrm{H}}(\tilde{\boldsymbol{u}} - \tilde{\boldsymbol{v}}) = \gamma m\}$ ▷ The naive algorithm is used.
  end if
  for $\tilde{\Theta}(q^{y\alpha_i m})$ times **do**
    Select $A_i \subset \{(\alpha_1 + \cdots + \alpha_{i-1})m + 1, \ldots, (\alpha_1 + \cdots + \alpha_i)m\}$ s.t. $|A_i| = \beta\alpha_i m$ u.a.r.
    $\mathcal{U}' \leftarrow \{\boldsymbol{u} \,|\, \boldsymbol{u} \in \tilde{\mathcal{U}}$ s.t. the number of coordinates in A_i equal to $x \in \mathbb{F}_q$ is $h_x\beta\alpha_i m\}$
    $\mathcal{V}' \leftarrow \{\boldsymbol{v} |\, \boldsymbol{v} \in \tilde{\mathcal{V}}$ s.t. the number of coordinates in A_i equal to $x \in \mathbb{F}_q$ is $h_x\beta\alpha_i m\}$ ▷ $\sum_{x\in\mathbb{F}_q} h_x = 1$
    if $|\mathcal{U}'|$ and $|\mathcal{V}'|$ are $\tilde{O}\left(q^{\left(\lambda\left(1-\sum_{j=1}^{i}\alpha_j\right)+\varepsilon/2\right)m}\right)$ **then**
      $\mathcal{C} \leftarrow \mathcal{C} \cup \mathrm{NNR}(\mathcal{U}', \mathcal{V}', m, t, \gamma, \lambda, \alpha_1, \ldots, \alpha_t, y, \varepsilon, i + 1)$
    end if
  end for
  return $\mathcal{C}$
end procedure

The time complexity of the May-Ozerov NN algorithm over $\mathbb{F}_q$ in Algorithm 1 is given by the following theorem. The proof of this theorem proceeds in the same way as the proof of Theorem 1 in [9].

Theorem 1. *Let q be any prime power. Let γ be any real such that $0 < \gamma < 1/2$. Let β be any real such that $0 < \beta < 1$. Let ε be any positive real and λ be any real such that*

$$\lambda \leq H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q(qh_x\beta) \tag{1}$$

with $\sum_{x\in\mathbb{F}_q} h_x = 1$ and $\gamma/q \leq h_x \leq \gamma/q + (1-\gamma)/(q\beta)$ for every $x \in \mathbb{F}_q$. Let

$$y = (1-\gamma)\left(H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q\left(\frac{qh_x-\gamma}{1-\gamma}\beta\right)\right) \quad . \tag{2}$$

Then, the May-Ozerov algorithm solves the (m, γ, λ)-NN problem over $\mathbb{F}_q$ with overwhelming probability in time $\tilde{O}(q^{(y+\varepsilon)m})$.

Proof. First, notice that, if $\gamma/q \leq h_x \leq \gamma/q + (1-\gamma)/(q\beta)$, then

$$0 \leq \frac{qh_x - \gamma}{1-\gamma}\beta \leq 1$$

and $0 < \gamma\beta < qh_x\beta < 1-(1-\beta)\gamma < 1$. It is also shown with elementary calculation that $\lambda < y$.

It will be shown in Lemma 1 that, for the for-loop from the line 7 to the line 13 in the procedure MO-NN, the probability that a solution randomized with a pair of $\boldsymbol{P}$ and $\boldsymbol{r}$ is included in $(\tilde{\mathcal{U}}, \tilde{\mathcal{V}})$ and given to the procedure NNR is $1/m^{O(1)}$.

For the for-loop from the line 19 to the line 26 in the procedure NNR, from Lemma 2, if the (randomized) solution is included in the pair of input lists $(\tilde{\mathcal{U}}, \tilde{\mathcal{V}})$ of the call of NNR with depth i, then the probability that the solution is also included in the lists $(\mathcal{U}', \mathcal{V}')$ of the input to the next call of NNR is $1/\tilde{\Theta}\left(q^{y\alpha_i m}\right)$ for a choice A_i of coordinates.

From Lemma 3, for $1 \leq i \leq t$, if the lists of the input to the call of NNR with depth $(i+1)$ includes the (randomized) solution, then their sizes are $\tilde{O}\left(q^{\left(\lambda\left(1-\sum_{j=1}^{i}\alpha_j\right)+\varepsilon/2\right)m}\right)$ with overwhelming probability for any $\varepsilon > 0$.

From the discussions above, the total number of the calls to NNR with depth i is $\tilde{O}\left(q^{\left(y\sum_{j=1}^{i-1}\alpha_j\right)m}\right)$ and each run of NNR with depth i takes $\tilde{O}\left(q^{y\alpha_i m+\left(\lambda\left(1-\sum_{j=1}^{i-1}\alpha_j\right)+\varepsilon/2\right)m}\right)$ time for $1 \leq i \leq t$. Thus, the total time complexity required to create the lists in the calls with depth i is

$$T_i = \tilde{O}\left(q^{\left(y\sum_{j=1}^{i}\alpha_j+\lambda\left(1-\sum_{j=1}^{i-1}\alpha_j\right)+\varepsilon/2\right)m}\right)$$

for $1 \leq i \leq t$. The total time complexity required by the calls with depth $(t+1)$ is $\tilde{O}\left(q^{(y+\varepsilon)m}\right)$.

Let us assume that $T_i = T_{i+1}$ for any i such that $1 \leq i \leq t-1$. Then,

$$y\sum_{j=1}^{i}\alpha_j + \lambda\left(1-\sum_{j=1}^{i-1}\alpha_j\right) + \varepsilon/2 = y\sum_{j=1}^{i+1}\alpha_j + \lambda\left(1-\sum_{j=1}^{i}\alpha_j\right) + \varepsilon/2$$

$$\alpha_{i+1} = (\lambda/y)\alpha_i \ .$$

Since $\lambda < y$ from Lemma 4 and $\sum_{j=1}^{t}\alpha_j = 1$, $\alpha_1 = \frac{1-\lambda/y}{1-(\lambda/y)^t}$ and

$$T_i = T_1 = \tilde{O}\left(q^{(y\alpha_1+\lambda+\varepsilon/2)m}\right)$$

for $1 \leq i \leq t$. Furthermore, if t is chosen to be

$$\left\lceil \frac{\log_2(y-\lambda+\varepsilon/2) - \log_2(\varepsilon/2)}{\log_2 y - \log_2 \lambda} \right\rceil \ ,$$

then $T_1 = \tilde{O}\left(q^{(y+\varepsilon)m}\right)$. It implies that the total time complexity of the May-Ozerov NN algorithm is $\tilde{O}\left(q^{(y+\varepsilon)m}\right)$. □

Lemma 1. *Let $(\mathcal{U}, \mathcal{V}, \gamma)$ be an instance of the (m, γ, λ)-NN problem with unknown solution $(\boldsymbol{u}^*, \boldsymbol{v}^*) \in \mathcal{U} \times \mathcal{V}$ such that $w_{\mathrm{H}}(\boldsymbol{u}^* - \boldsymbol{v}^*) = \gamma m$. Let $\boldsymbol{z}^* = \boldsymbol{u}^* - \boldsymbol{v}^*$. Let t be a constant integer. Let $\alpha_1, \alpha_2, \ldots, \alpha_t$ be positive reals satisfying $\alpha_1 + \cdots + \alpha_t = 1$. Let $\boldsymbol{P}$ be a permutation matrix chosen uniformly at random from $\{0,1\}^{m\times m}$. Let $\boldsymbol{r} \in \mathbb{F}_q^m$ be a vector chosen uniformly at random such that $\boldsymbol{r} = (\boldsymbol{r}_1, \ldots, \boldsymbol{r}_t)$ with $\boldsymbol{r}_i \in \mathbb{F}_q^{\alpha_i m}$ balanced for every $1 \leq i \leq t$. Let*

$$\tilde{\boldsymbol{u}}^* = \boldsymbol{P}\boldsymbol{u}^* + \boldsymbol{r} = (\tilde{\boldsymbol{u}}_1^*, \tilde{\boldsymbol{u}}_2^*, \ldots, \tilde{\boldsymbol{u}}_t^*) \ ,$$
$$\tilde{\boldsymbol{v}}^* = \boldsymbol{P}\boldsymbol{v}^* + \boldsymbol{r} = (\tilde{\boldsymbol{v}}_1^*, \tilde{\boldsymbol{v}}_2^*, \ldots, \tilde{\boldsymbol{v}}_t^*) \ ,$$
$$\tilde{\boldsymbol{z}}^* = \tilde{\boldsymbol{u}}^* - \tilde{\boldsymbol{v}}^* = P\boldsymbol{z}^* = (\tilde{\boldsymbol{z}}_1^*, \tilde{\boldsymbol{z}}_2^*, \ldots, \tilde{\boldsymbol{z}}_t^*) \ ,$$

where $\tilde{\boldsymbol{u}}_i^* \in \mathbb{F}_q^{\alpha_i m}$, $\tilde{\boldsymbol{v}}_i^* \in \mathbb{F}_q^{\alpha_i m}$ *and* $\tilde{\boldsymbol{z}}_i^* \in \mathbb{F}_q^{\alpha_i m}$ *for every* $1 \le i \le t$. *Then, the probability that both* $\tilde{\boldsymbol{u}}_i^*$ *and* $\tilde{\boldsymbol{v}}_i^*$ *are balanced and* $w_{\mathrm{H}}(\tilde{\boldsymbol{z}}_i^*) = \gamma\alpha_i m$ *for every* $1 \le i \le t$ *is*

$$1 \Big/ O\left(m^{\frac{(q-1)^2(q+1)t+t-1}{2}}\right) \quad .$$

Proof. For $\tilde{\boldsymbol{u}}_i^*$ and $\tilde{\boldsymbol{v}}_i^*$, let $\mathtt{Bal}_i$ be the event that both of $\tilde{\boldsymbol{u}}_i^*$ and $\tilde{\boldsymbol{v}}_i^*$ are balanced. Let $\tilde{\boldsymbol{u}}_i^* - \boldsymbol{r}_i = (\hat{u}_{i,1}^*, \hat{u}_{i,2}^*, \ldots, \hat{u}_{i,\alpha_i m}^*)$ and $\tilde{\boldsymbol{v}}_i^* - \boldsymbol{r}_i = (\hat{v}_{i,1}^*, \hat{v}_{i,2}^*, \ldots, \hat{v}_{i,\alpha_i m}^*)$. Let $\hat{S}_{x,y} = \{j \mid (\hat{u}_{i,j}^* = x) \wedge (\hat{v}_{i,j}^* = y)\}$ for $(x, y) \in \mathbb{F}_q^2$. Then, $\mathtt{Bal}_i$ occurs if $\boldsymbol{r}_i$ is balanced on the coordinates in $\hat{S}_{x,y}$ for every $(x, y) \in \mathbb{F}_q^2$. Thus,

$$\begin{aligned}
\Pr[\mathtt{Bal}_i] &\ge \binom{\alpha_i m}{\alpha_i m/q, \ldots, \alpha_i m/q}^{-1} \prod_{(x,y)\in\mathbb{F}_q^2} \binom{|\hat{S}_{x,y}|}{|\hat{S}_{x,y}|/q, \ldots, |\hat{S}_{x,y}|/q} \\
&\approx \left(\frac{q^q}{(2\pi)^{q-1}}\right)^{\frac{(q-1)(q+1)}{2}} \left(\alpha_i m \Big/ \prod_{(x,y)\in\mathbb{F}_q^2} |\hat{S}_{x,y}|\right)^{\frac{q-1}{2}} = 1 \Big/ O\left(m^{\frac{(q-1)^2(q+1)}{2}}\right) \quad .
\end{aligned}$$

For $\tilde{\boldsymbol{z}}^*$, since $\boldsymbol{P}$ is chosen uniformly at random,

$$\begin{aligned}
\Pr\left[\bigwedge_{i=1}^{t} (w_{\mathrm{H}}(\tilde{\boldsymbol{z}}_i^*) = \gamma\alpha_i m)\right] &= \binom{m}{\gamma m}^{-1} \prod_{i=1}^{t} \binom{\alpha_i m}{\gamma\alpha_i m} \\
&\approx (2\pi\gamma(1-\gamma))^{-\frac{t-1}{2}} (\alpha_1 \cdots \alpha_t)^{-\frac{1}{2}} m^{-\frac{t-1}{2}} = 1 \Big/ \Theta\left(m^{\frac{t-1}{2}}\right) \quad .
\end{aligned}$$

Since $\boldsymbol{P}$ and $\boldsymbol{r}$ are independent of each other,

$$\Pr\left[\bigwedge_{i=1}^{t} ((w_{\mathrm{H}}(\hat{\boldsymbol{z}}_i^*) = \gamma\alpha_i m) \wedge \mathtt{Bal}_i)\right] = 1 \Big/ O\left(m^{\frac{(q-1)^2(q+1)t+t-1}{2}}\right) \quad .$$

□

From Lemma 1, with $O\left(m^{\frac{(q-1)^2(q+1)t+t-1}{2}}\right)$ executions of the for-loop from the line 7 to the line 13, the randomized unknown solution satisfying the conditions in Lemma 1 is given to the procedure NNR with overwhelming probability. Notice that the condition $w_{\mathrm{H}}(\tilde{\boldsymbol{z}}_i^*) = \gamma\alpha_i m$ for $1 \le i \le t$ cannot be checked since the solution is unknown. The proof of Lemma 1 validates the algorithm only if each piece of vectors has at least q^3 coordinates.

Lemma 2. *For a recursive call to the procedure NNR in the May-Ozerov algorithm, suppose that, for input* $(\tilde{\mathcal{U}}, \tilde{\mathcal{V}}, m, t, \gamma, \lambda, \alpha_1, \ldots, \alpha_t, y, \varepsilon, i)$, $\tilde{\mathcal{U}} \times \tilde{\mathcal{V}}$ *includes a (randomized) unknown solution* $(\tilde{\boldsymbol{u}}^*, \tilde{\boldsymbol{v}}^*)$. *Then, the probability that the input* $\mathcal{U}' \times \mathcal{V}'$ *to the next call also includes* $(\tilde{\boldsymbol{u}}^*, \tilde{\boldsymbol{v}}^*)$ *is* $1/\tilde{O}(q^{y\alpha_i m})$ *if* A_i *is chosen uniformly at random.*

Proof. From Lemma 1 and its proof, it is assumed that $\tilde{\boldsymbol{u}}_i^*$ and $\tilde{\boldsymbol{v}}_i^*$ satisfy the conditions in Lemma 1 and that $\boldsymbol{r}_i$ is balanced on the coordinates in $\hat{S}_{x,y}$ for every $(x, y) \in \mathbb{F}_q^2$. Let $\tilde{\boldsymbol{u}}_i^* = (\tilde{u}_{i,1}^*, \tilde{u}_{i,2}^*, \ldots, \tilde{u}_{i,\alpha_i m}^*)$ and $\tilde{\boldsymbol{v}}_i^* = (\tilde{v}_{i,1}^*, \tilde{v}_{i,2}^*, \ldots, \tilde{v}_{i,\alpha_i m}^*)$. Let $\tilde{S}_{x,y} = \{j \mid (\tilde{u}_{i,j}^* = x) \wedge (\tilde{v}_{i,j}^* = y)\}$ for $(x, y) \in \mathbb{F}_q^2$.

Since $w_{\mathrm{H}}(\tilde{\boldsymbol{u}}_i^* - \tilde{\boldsymbol{v}}_i^*) = \gamma\alpha_i m$ and $\boldsymbol{r}_i$ is balanced on the coordinates in $\hat{S}_{x,x}$ for every $x \in \mathbb{F}_q$, $\left|\tilde{S}_{x,x}\right| = (1-\gamma)\alpha_i m/q$ and

$$\sum_{y\in\mathbb{F}_q\setminus\{x\}} \left|\tilde{S}_{x,y}\right| = \sum_{y\in\mathbb{F}_q\setminus\{x\}} \left|\tilde{S}_{y,x}\right| = \frac{\alpha_i m}{q} - \frac{(1-\gamma)\alpha_i m}{q} = \frac{\gamma\alpha_i m}{q}$$

for every $x \in \mathbb{F}_q$.

For $x \in \mathbb{F}_q$, let h_x be positive reals such that $\sum_{x\in\mathbb{F}_q} h_x = 1$. The number of x in coordinates of $\tilde{\boldsymbol{u}}^*$ in A_i is $h_x\beta\alpha_i m$ if $\left|A_i \cap \tilde{S}_{x,y}\right| = \beta\left|\tilde{S}_{x,y}\right|$ for every $y \in \mathbb{F}_q \setminus \{x\}$ and $\left|A_i \cap \tilde{S}_{x,x}\right| = h_x\beta\alpha_i m - \sum_{y\in\mathbb{F}_q\setminus\{x\}} \beta\left|\tilde{S}_{x,y}\right|$. Thus,

$$
\begin{aligned}
&\Pr\left[\bigwedge_{x\in\mathbb{F}_q}\left(\left|A_i \cap \bigcup_{y\in\mathbb{F}_q}\tilde{S}_{x,y}\right| = h_x\beta\alpha_i m\right)\right] \\
&\geq \Pr\left[\bigwedge_{x\in\mathbb{F}_q}\left(\left(\left|A_i\cap\tilde{S}_{x,x}\right| = h_x\beta\alpha_i m - \sum_{y\in\mathbb{F}_q\setminus\{x\}}\beta\left|\tilde{S}_{x,y}\right|\right) \wedge \bigwedge_{y\in\mathbb{F}_q\setminus\{x\}}\left(\left|A_i\cap\tilde{S}_{x,y}\right| = \beta\left|\tilde{S}_{x,y}\right|\right)\right)\right] \\
&= \binom{\alpha_i m}{\beta\alpha_i m}^{-1}\prod_{x\in\mathbb{F}_q}\left(\binom{|\tilde{S}_{x,x}|}{h_x\beta\alpha_i m - \sum_{y\in\mathbb{F}_q\setminus\{x\}}\beta|\tilde{S}_{x,y}|}\prod_{y\in\mathbb{F}_q\setminus\{x\}}\binom{|\tilde{S}_{x,y}|}{\beta|\tilde{S}_{x,y}|}\right) \\
&= 1\Big/\tilde{\Theta}\left(2^{(1-\gamma)\left(H(\beta)-\frac{1}{q}\sum_{x\in\mathbb{F}_q}H\left(\frac{qh_x-\gamma}{1-\gamma}\beta\right)\right)\alpha_i m}\right) \\
&= 1/\tilde{\Theta}\left(q^{y\alpha_i m}\right) \ ,
\end{aligned}
$$

where

$$
y = (1-\gamma)\left(H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q\left(\frac{qh_x-\gamma}{1-\gamma}\beta\right)\right) \ .
$$

Since $0 \leq \frac{qh_x-\gamma}{1-\gamma}\beta \leq 1$, $\gamma/q \leq h_x \leq \gamma/q + (1-\gamma)/(q\beta)$ for every $x \in \mathbb{F}_q$.

Notice that, if

- $\left|A_i \cap \tilde{S}_{x,y}\right| = \beta\left|\tilde{S}_{x,y}\right|$ for every $(x,y) \in \mathbb{F}_q^2$ such that $x \neq y$, and
- $\left|A_i \cap \tilde{S}_{x,x}\right| = h_x\beta\alpha_i m - \sum_{y\in\mathbb{F}_q\setminus\{x\}}\beta\left|\tilde{S}_{x,y}\right|$ for every $x \in \mathbb{F}_q$,

the number of x in coordinates of $\tilde{\boldsymbol{v}}^*$ in A_i is also $h_x\beta\alpha_i m$ for every $x \in \mathbb{F}_q$. □

Lemma 3. *For the (m,γ,λ)-NN problem over $\mathbb{F}_q$, suppose that*

$$
\lambda \leq H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q(qh_x\beta) \ ,
$$

where $\sum_{x\in\mathbb{F}_q} h_x = 1$ and $\gamma/q \leq h_x \leq \gamma/q + (1-\gamma)/(q\beta)$ for every $x \in \mathbb{F}_q$. For a recursive call to NNR with depth i in the May-Ozerov algorithm, if the input lists $\tilde{\mathcal{U}} \times \tilde{\mathcal{V}}$ include a (randomized) unknown solution, then the probability that $|\mathcal{U}'|$ and $|\mathcal{V}'|$ are $\tilde{O}\left(q^{\left(\lambda\left(1-\sum_{j=1}^{i}\alpha_j\right)+\varepsilon/2\right)m}\right)$ is at least $1-1/q^{\varepsilon m}$ for the input lists $\mathcal{U}'$ and $\mathcal{V}'$ to the next call which include the (randomized) unknown solution.

Proof. Let us call a sequence of calls to NNR from the root to a leaf a good computation path if the sequence finally outputs an unknown solution. For $1 \leq i \leq t$, let $\mathcal{U}_i^*$ and $\mathcal{V}_i^*$ be the lists computed at depth i on the good computation path. $|\mathcal{U}_i^*|$ is evaluated below. $|\mathcal{V}_i^*|$ can be evaluated in the same way. Notice that h_x's are denoted by $h_0, h_1, \ldots, h_{q-1}$ in this proof.

Let $X_i^{\boldsymbol{u}}$ be a random variable for $\boldsymbol{u} \in \mathcal{U}$ such that

$$
X_i^{\boldsymbol{u}} = \begin{cases} 1 & \text{if } \tilde{\boldsymbol{u}} \in \mathcal{U}_1^* \wedge \tilde{\boldsymbol{u}} \in \mathcal{U}_2^* \wedge \cdots \wedge \tilde{\boldsymbol{u}} \in \mathcal{U}_i^* \\ 0 & \text{otherwise.} \end{cases}
$$

Then, for every $\boldsymbol{u} \in \mathcal{U} \setminus \{\boldsymbol{u}^*\}$,

$$
\begin{aligned}
&\Pr[X_i^{\boldsymbol{u}} = 1] \\
&= \prod_{j=1}^{i} \binom{\beta\alpha_j m}{h_0\beta\alpha_j m, \ldots, h_{q-1}\beta\alpha_j m} \binom{(1-\beta)\alpha_j m}{\left(\frac{1}{q} - h_0\beta\right)\alpha_j m, \ldots, \left(\frac{1}{q} - h_{q-1}\beta\right)\alpha_j m} \binom{\alpha_j m}{\frac{\alpha_j m}{q}, \ldots, \frac{\alpha_j m}{q}}^{-1} \\
&= 1 \Big/ \tilde{\Theta}\left(q^{\left(1-(1-\beta)\log_q\left(\frac{1}{\beta}-1\right)+\beta\sum_{x=0}^{q-1}\left(h_x \log_q h_x + \left(\frac{1}{q\beta}-h_x\right)\log_q\left(\frac{1}{q\beta}-h_x\right)\right)\right)\sum_{j=1}^{i}\alpha_j m}\right) \\
&= 1 \Big/ \tilde{\Theta}\left(q^{\left(H_q(\beta)-\frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q(qh_x\beta)\right)\sum_{j=1}^{i}\alpha_j m}\right) \\
&\le 1 \Big/ \tilde{\Theta}\left(q^{\lambda\sum_{j=1}^{i}\alpha_j m}\right) \quad ,
\end{aligned}
$$

where $h_0, h_1, \ldots, h_{q-1}$ are chosen to satisfy

$$\lambda \le H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q(qh_x\beta) \ .$$

Let $X_i = \sum_{\boldsymbol{u}\in\mathcal{U}} X_i^{\boldsymbol{u}}$. Then,

$$\mathrm{E}[X_i] \le 1 + (q^{\lambda m} - 1) \Big/ \tilde{\Theta}\left(q^{\lambda\sum_{j=1}^{i}\alpha_j m}\right) = \tilde{\Theta}\left(q^{\lambda\left(1-\sum_{j=1}^{i}\alpha_j\right)m}\right) \ .$$

From Chebyshev's inequality,

$$\Pr\left[|X_i - \mathrm{E}[X_i]| \ge q^{\frac{\varepsilon}{2}m}\mathrm{E}[X_i]\right] \le \frac{\mathrm{V}[X_i]}{q^{\varepsilon m}(\mathrm{E}[X_i])^2} \le \frac{1}{q^{\varepsilon m}\mathrm{E}[X_i]} \le \frac{1}{q^{\varepsilon m}} \ ,$$

where, since $X_i^{\boldsymbol{u}}$'s are pairwise independent and $X_i^{\boldsymbol{u}}$ is 0 or 1,

$$\mathrm{V}[X_i] = \mathrm{V}\left[\sum_{\boldsymbol{u}\in\mathcal{U}} X_i^{\boldsymbol{u}}\right] = \sum_{\boldsymbol{u}\in\mathcal{U}} \mathrm{V}[X_i^{\boldsymbol{u}}] = \sum_{\boldsymbol{u}\in\mathcal{U}} \left(\mathrm{E}[(X_i^{\boldsymbol{u}})^2] - \mathrm{E}[X_i^{\boldsymbol{u}}]^2\right) \le \sum_{\boldsymbol{u}\in\mathcal{U}} \mathrm{E}[X_i^{\boldsymbol{u}}] = \mathrm{E}[X_i] \ .$$

Thus, with probability at least $1 - 1/q^{\varepsilon m}$, $|\mathcal{U}_i^*| = \tilde{O}\left(q^{\lambda\left(1-\sum_{j=1}^{i}\alpha_j\right)m + \frac{\varepsilon}{2}m}\right)$. □

Lemma 4. *Let q be any prime power. Let γ be any real such that $0 < \gamma < 1/2$. Let β be any real such that $0 < \beta < 1$. Then,*

$$H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q(qh_x\beta) \le (1-\gamma)\left(H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q\left(\frac{qh_x - \gamma}{1-\gamma}\beta\right)\right)$$

for all h_x's such that $\sum_{x\in\mathbb{F}_q} h_x = 1$ and $\gamma/q \le h_x \le \gamma/q + (1-\gamma)/(q\beta)$ for every $x \in \mathbb{F}_q$. The equality is satisfied iff $h_x = 1/q$ for every $x \in \mathbb{F}_q$. In this case, both sides are equal to 0.

A proof of Lemma 4 is given in Apendix A.

4 Stern ISD Using May-Ozerov NN Algorithm over $\mathbb{F}_q$

May and Ozerov applied their algorithm for the nearest-neighbor problem to the Stern ISD for linear codes over $\mathbb{F}_2$ [9]. It is quite straightforward to generalize it for linear codes over other finite fields with the algorithm presented in the previous section. The generalized decoding algorithm is given in Algorithm 2. As was mentioned earlier, the bounded distance decoding is considered. It is also assumed that, for a given instance $(n, k, \boldsymbol{H}, \boldsymbol{x})$, the distance d satisfies $k/n = 1 - H_q(d/n)$ and the distance between $\boldsymbol{x}$ and the closest codeword is $w = \lfloor (d-1)/2 \rfloor$.

Algorithm 2 Stern ISD with May-Ozerov Nearest-Neighbor Algorithm over $\mathbb{F}_q$

1: **procedure** ISD$(n, k, \boldsymbol{H}, \boldsymbol{x})$ ▷ $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\boldsymbol{x} \in \mathbb{F}_q^n$
2: $\quad \boldsymbol{s} \leftarrow \boldsymbol{H}\boldsymbol{x}$
3: $\quad d \leftarrow H_q^{-1}(1-k/n)\cdot n$
4: $\quad w \leftarrow \lfloor (d-1)/2 \rfloor$
5: $\quad$ Select p ▷ $\max\{1, w+k-n\} \le p \le \min\{k, w\}$
6: $\quad$ **repeat**
7: $\quad\quad$ **repeat**
8: $\quad\quad\quad$ Select a permutation matrix $\boldsymbol{P} \in \{0,1\}^{n\times n}$ u.a.r.
9: $\quad\quad\quad$ $(\ \cdot\ \ \boldsymbol{Q}) \leftarrow \boldsymbol{H}\boldsymbol{P}$
10: $\quad\quad$ **until** $\boldsymbol{Q}$ is non-singular
11: $\quad\quad$ $\tilde{\boldsymbol{H}} \leftarrow \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{P}$
12: $\quad\quad$ $\tilde{\boldsymbol{s}} \leftarrow \boldsymbol{Q}^{-1}\boldsymbol{s}$
13: $\quad\quad$ $\mathcal{U} \leftarrow \{\boldsymbol{u} \mid \boldsymbol{u} = \tilde{\boldsymbol{H}}\boldsymbol{e}_1$ for $\boldsymbol{e}_1 \in \mathbb{F}_q^{k/2} \times \{0\}^{k/2} \times \{0\}^{n-k}$ s.t. $w_{\mathrm{H}}(\boldsymbol{e}_1) = p/2\}$
14: $\quad\quad$ $\mathcal{V} \leftarrow \{\boldsymbol{v} \mid \boldsymbol{v} = \tilde{\boldsymbol{H}}\boldsymbol{e}_2 + \tilde{\boldsymbol{s}}$ for $\boldsymbol{e}_2 \in \{0\}^{k/2} \times \mathbb{F}_q^{k/2} \times \{0\}^{n-k}$ s.t. $w_{\mathrm{H}}(\boldsymbol{e}_2) = p/2\}$
15: $\quad\quad$ $\mathcal{C} \leftarrow$ MO-NN$(\mathcal{U}, \mathcal{V}, (w-p)/(n-k))$ ▷ Run the May-Ozerov NN algorithm over $\mathbb{F}_q$
16: $\quad$ **until** there exists $(\boldsymbol{u}^*, \boldsymbol{v}^*) \in \mathcal{C}$ s.t. $w_{\mathrm{H}}(\boldsymbol{u}^* - \boldsymbol{v}^*) = w - p$
17: $\quad$ **return** $\boldsymbol{P}(\boldsymbol{e}_1^* - \boldsymbol{e}_2^* - (0^k \| (\boldsymbol{u}^* - \boldsymbol{v}^*)))$ ▷ $\boldsymbol{u}^* = \tilde{\boldsymbol{H}}\boldsymbol{e}_1^*$ and $\boldsymbol{v}^* = \tilde{\boldsymbol{H}}\boldsymbol{e}_2^* + \tilde{\boldsymbol{s}}$
18: **end procedure**

Theorem 2. *For any $\varepsilon > 0$, the Stern ISD with the May-Ozerov NN algorithm solves the decoding problem of random $[n, k]$ linear codes over $\mathbb{F}_q$ with overwhelming probability in time*

$$\min_{p,\beta,\{h_x \mid x\in\mathbb{F}_q\}} \tilde{O}\left(q^{g(q,n,k,w,p,\beta,\{h_x \mid x\in\mathbb{F}_q\},\varepsilon)}\right) \ ,$$

where

$$g\left(q,n,k,w,p,\beta,\{h_x \mid x\in\mathbb{F}_q\},\varepsilon\right) = \\ (\log_q 2)\left(nH\left(\frac{w}{n}\right) - kH\left(\frac{p}{k}\right) - (n-k)H\left(\frac{w-p}{n-k}\right)\right) + (y+\varepsilon)(n-k)$$

and

$$y = (1-\gamma)\left(H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q\left(\frac{qh_x - \gamma}{1-\gamma}\beta\right)\right)$$

with $\gamma = (w-p)/(n-k)$. The conditions on p, β and $\{h_x \mid x \in \mathbb{F}_q\}$ for minimization are

- $\max\{1, w+k-n\} \le p \le \min\{k, w\}$,
- $0 < \beta < 1$,
- $\displaystyle\sum_{x\in\mathbb{F}_q} h_x = 1$,
- $\gamma/q \le h_x \le \gamma/q + (1-\gamma)/(q\beta)$, *and*
- $\dfrac{(k/2)H_q(p/k)}{n-k} < H_q(\beta) - \dfrac{1}{q}\displaystyle\sum_{x\in\mathbb{F}_q} H_q(qh_x\beta)$.

Proof. The expected number of iterations of the repeat-until-loop from the line 6 to the line 16 in Algorithm 2 is

$$\tilde{O}\left(\frac{\binom{n}{w}}{\binom{k/2}{p/2}^2\binom{n-k}{w-p}}\right) = \tilde{O}\left(2^{(nH(w/n) - kH(p/k) - (n-k)H((w-p)/(n-k)))}\right) \ .$$

The sizes of the lists $\mathcal{U}$ and $\mathcal{V}$ are

$$|\mathcal{U}| = |\mathcal{V}| = \binom{k/2}{p/2} (q-1)^{p/2} = \tilde{O}\left(q^{(k/2)H_q(p/k)}\right) \ .$$

Thus, MO-NN is given an instance of the (m, γ, λ)-NN problem with

$$m = n - k \ , \quad \gamma = \frac{w-p}{n-k} \ , \quad \lambda = \frac{(k/2)H_q(p/k)}{n-k} \ .$$

Lemma 3 only requires

$$\lambda \leq H_q(\beta) - \frac{1}{q} \sum_{x \in \mathbb{F}_q} H_q(q h_x \beta) \ .$$

On the other hand,

$$H_q(\beta) - \frac{1}{q} \sum_{x \in \mathbb{F}_q} H_q(q h_x \beta) < y \ .$$

Thus, for the minimization of the time complexity, it is assumed that

$$\lambda = H_q(\beta) - \frac{1}{q} \sum_{x \in \mathbb{F}_q} H_q(q h_x \beta)$$

without loss of generality. □

5 Numerical Analysis of Time Complexity

Some numerical analyses are given to the asymptotic time complexity of the Stern ISD using the May-Ozerov NN algorithm over $\mathbb{F}_q$.

For the time complexity of the Stern ISD with May-Ozerov NN algorithm over $\mathbb{F}_q$, let

$$T(q, n, k, w) = \min_{p, \beta, \{h_x \,|\, x \in \mathbb{F}_q\}} q^{g(q,n,k,w,p,\beta,\{h_x \,|\, x \in \mathbb{F}_q\},\varepsilon)} \ .$$

Then, $\lim_{n\to\infty} \frac{1}{n} \log_q T(q, n, k, w)$ is a function of q and $R = k/n$. Let us denote it by $f(q, R)$. The asymptotic time complexity is evaluated with $f(q, R)$. Since $\varepsilon > 0$ is arbitrary from Theorem 2, it is neglected in the analysis given below.

To obtain the values of $f(q, R)$, the numerical optimization problem given in Theorem 2 is solved for $q = 2, 3, 4$. For $q = 3, 4$, the optimal values are obtained for h_x's such that all but one of them have the same value. Thus, for some larger values of q, the optimization problem is solved on the assumption that all but one of h_x's are equal to each other.

The curves of $f(q, R)$ for $q = 2, 3, 4, 5, 7, 8, 11$ are given in Figure 1. $f(q, R)$ gets smaller as q gets larger.

Table 1 presents the asymptotic time complexity of the worst cases for bounded distance decoding. In this table, Stern-MO represents the Stern ISD with the May-Ozerov NN algorithm, and Stern represents the Stern ISD given in Algorithm 3. $f_{\mathrm{S}}(q, R)$ is defined for the Stern ISD similarly to $f(q, R)$. The results for $q = 2$ are consistent with the results by May and Ozerov in [9]. It is shown that, in this analysis, the Stern-MO algorithm outperforms the Stern algorithm only over $\mathbb{F}_2$. For $q \geq 5$, as q gets larger, the degradation of the Stern-MO algorithm gets smaller.

6 Conclusion

The paper have shown the generalization of the May-Ozerov NN algorithm over $\mathbb{F}_q$ with any prime power q. The complexity analysis suggests that the May-Ozerov NN algorithm over $\mathbb{F}_q$ may not be practical even for small prime $q \geq 3$ due to the huge polynomial which does not appear in the $\tilde{O}$ notation of its time complexity. It is an open problem if more rigorous analysis or some other generalization over $\mathbb{F}_q$ reduces the time complexity. It is also left as future work to analyze the complexity of the BJMM information set decoding with the May-Ozerov NN algorithm over $\mathbb{F}_q$.

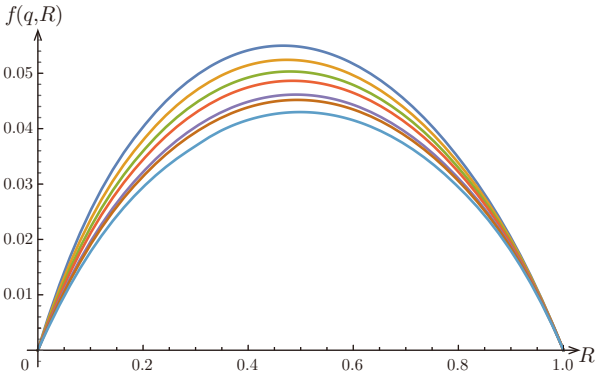

Fig. 1: Asymptotic time complexity of the Stern ISD with the May-Ozerov NN algorithm over $\mathbb{F}_q$. $q = 2, 3, 4, 5, 7, 8, 11$ in the decreasing order.

Table 1: Asymptotic time complexity of worst cases for bounded distance decoding. $\Delta = f(q, R_{\mathrm{w}}) - f_{\mathrm{S}}(q, R'_{\mathrm{w}})$. For Stern-MO, all but one of h_x's are equal to h.

	Stern-MO					Stern		
q	$f(q, R_{\mathrm{w}})$	R_{w}	p/n	β	h	$f_{\mathrm{S}}(q, R'_{\mathrm{w}})$	R'_{w}	Δ
2	.05498	.4663	.003848	.4998	.3981	.05563	.4655	−.00065
3	.05242	.4736	.002979	.1792	.2322	.05217	.4742	.00025
4	.05032	.4796	.002201	.0932	.1644	.04987	.4801	.00045
5	.04864	.4843	.001704	.0593	.1279	.04815	.4844	.00049
7	.04614	.4909	.001164	.0326	.0893	.04571	.4907	.00043
8	.04519	.4933	.001006	.0263	.0778	.04478	.4931	.00041
11	.04299	.4989	.000727	.0166	.0563	.04266	.4985	.00033

Acknowledgments

We would like to thank one of the anonymous reviewers of PQCrypto 2016 for his/her insightful comments. Any errors or mistakes remain the sole responsibility of the authors. This work was partially supported by JSPS KAKENHI Grant Number 25330152.

References

1. Becker, A., Joux, A., May, A., Meurer, A.: Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In: Pointcheval, D., Johansson, T. (eds.) Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings. Lecture Notes in Computer Science, vol. 7237, pp. 520–536. Springer (2012), http://dx.doi.org/10.1007/978-3-642-29011-4_31
2. Bernstein, D.J., Lange, T., Peters, C.: Smaller decoding exponents: Ball-collision decoding. In: Rogaway, P. (ed.) Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings. Lecture Notes in Computer Science, vol. 6841, pp. 743–760. Springer (2011), http://dx.doi.org/10.1007/978-3-642-22792-9_42
3. Coffey, J.T., Goodman, R.M.: Any code of which we cannot think is good. IEEE Transactions on Information Theory 36(6), 1453–1461 (1990), http://dx.doi.org/10.1109/18.59944
4. Coffey, J.T., Goodman, R.M.: The complexity of information set decoding. IEEE Transactions on Information Theory 36(5), 1031–1037 (1990), http://dx.doi.org/10.1109/18.57202
5. Finiasz, M., Sendrier, N.: Security bounds for the design of code-based cryptosystems. In: Matsui, M. (ed.) Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5912, pp. 88–105. Springer (2009), http://dx.doi.org/10.1007/978-3-642-10366-7_6
6. Howgrave-Graham, N., Joux, A.: New generic algorithms for hard knapsacks. In: Gilbert, H. (ed.) Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6110, pp. 235–256. Springer (2010), http://dx.doi.org/10.1007/978-3-642-13190-5_12

Algorithm 3 Stern ISD over $\mathbb{F}_q$

procedure ISD$(n, k, \boldsymbol{H}, \boldsymbol{x})$ ▷ $\boldsymbol{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\boldsymbol{x} \in \mathbb{F}_q^n$
  $\boldsymbol{s} \leftarrow \boldsymbol{H}\boldsymbol{x}$
  $d \leftarrow H_q^{-1}(1 - k/n) \cdot n$
  $w \leftarrow \lfloor (d-1)/2 \rfloor$
  Select p and ℓ ▷ $0 \le \ell \le n-k$ and $\max\{0, k+w+\ell-n\} < p < \min\{k, w\}$
  repeat
    repeat
      Select a permutation matrix $\boldsymbol{P} \in \{0,1\}^{n\times n}$ u.a.r.
      $(\ \cdot\ \ \boldsymbol{Q}) \leftarrow \boldsymbol{H}\boldsymbol{P}$
    until $\boldsymbol{Q}$ is non-singular
    $\tilde{\boldsymbol{H}} \leftarrow \boldsymbol{Q}^{-1}\boldsymbol{H}\boldsymbol{P}$
    $\tilde{\boldsymbol{s}} \leftarrow \boldsymbol{Q}^{-1}\boldsymbol{s}$
    $\mathcal{U} \leftarrow \{\boldsymbol{u} \,|\, \boldsymbol{u} = \tilde{\boldsymbol{H}}\boldsymbol{e}_1$ for $\boldsymbol{e}_1 \in \mathbb{F}_q^{k/2} \times \{0\}^{k/2} \times \{0\}^{n-k}$ s.t. $w_{\mathrm{H}}(\boldsymbol{e}_1) = p/2\}$
    $\mathcal{V} \leftarrow \{\boldsymbol{v} \,|\, \boldsymbol{v} = \tilde{\boldsymbol{H}}\boldsymbol{e}_2 + \tilde{\boldsymbol{s}}$ for $\boldsymbol{e}_2 \in \{0\}^{k/2} \times \mathbb{F}_q^{k/2} \times \{0\}^{n-k}$ s.t. $w_{\mathrm{H}}(\boldsymbol{e}_2) = p/2\}$
    sort the vectors in $\mathcal{U}$ with respect to the last ℓ coordinates
    sort the vectors in $\mathcal{V}$ with respect to the last ℓ coordinates
    for all $(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{U} \times \mathcal{V}$ s.t. $\boldsymbol{u}$ and $\boldsymbol{v}$ are equal in the last ℓ coordinates **do**
      check if $w_{\mathrm{H}}(\boldsymbol{u} - \boldsymbol{v}) = w - p$
    end for
  until there exists $(\boldsymbol{u}^*, \boldsymbol{v}^*) \in \mathcal{U} \times \mathcal{V}$ s.t. $w_{\mathrm{H}}(\boldsymbol{u}^* - \boldsymbol{v}^*) = w - p$
  return $\boldsymbol{P}(\boldsymbol{e}_1^* - \boldsymbol{e}_2^* - (0^k \| (\boldsymbol{u}^* - \boldsymbol{v}^*)))$ ▷ $\boldsymbol{u}^* = \tilde{\boldsymbol{H}}\boldsymbol{e}_1^*$ and $\boldsymbol{v}^* = \tilde{\boldsymbol{H}}\boldsymbol{e}_2^* + \tilde{\boldsymbol{s}}$
end procedure

7. Lee, P.J., Brickell, E.F.: An observation on the security of McEliece's public-key cryptosystem. In: Günther, C.G. (ed.) Advances in Cryptology - EUROCRYPT '88, Workshop on the Theory and Application of of Cryptographic Techniques, Davos, Switzerland, May 25-27, 1988, Proceedings. Lecture Notes in Computer Science, vol. 330, pp. 275–280. Springer (1988), http://dx.doi.org/10.1007/3-540-45961-8_25
8. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings. Lecture Notes in Computer Science, vol. 7073, pp. 107–124. Springer (2011), http://dx.doi.org/10.1007/978-3-642-25385-0_6
9. May, A., Ozerov, I.: On computing nearest neighbors with applications to decoding of binary linear codes. In: Oswald, E., Fischlin, M. (eds.) Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I. Lecture Notes in Computer Science, vol. 9056, pp. 203–228. Springer (2015), http://dx.doi.org/10.1007/978-3-662-46800-5_9
10. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Jet Propulsion Laboratory DSN Progress Report 4244 (1978)
11. Meurer, A.: A Coding-Theoretic Approach to Cryptanalysis. Ph.D. thesis, Ruhr-University Bochum (2012)
12. Peters, C.: Information-set decoding for linear codes over $\mathbb{F}_q$. In: Sendrier, N. (ed.) Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6061, pp. 81–94. Springer (2010), http://dx.doi.org/10.1007/978-3-642-12929-2_7
13. Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions on Information Theory 8(5), 5–9 (1962), http://dx.doi.org/10.1109/TIT.1962.1057777
14. Stern, J.: A method for finding codewords of small weight. In: Cohen, G.D., Wolfmann, J. (eds.) Coding Theory and Applications, 3rd International Colloquium, Toulon, France, November 2-4, 1988, Proceedings. Lecture Notes in Computer Science, vol. 388, pp. 106–113. Springer (1988)

A Proof of Lemma 4

Let

$$G(h) = H_q(qh\beta) - (1-\gamma)H_q\left(\frac{qh-\gamma}{1-\gamma}\beta\right) - \gamma H_q(\beta) \ .$$

Then,

$$\begin{aligned}
&(1-\gamma)\left(H_q(\beta)-\frac{1}{q}\sum_{x\in\mathbb{F}_q}H_q\left(\frac{qh_x-\gamma}{1-\gamma}\beta\right)\right)-\left(H_q(\beta)-\frac{1}{q}\sum_{x\in\mathbb{F}_q}H_q(qh_x\beta)\right)\\
&\quad=-\gamma H_q(\beta)-(1-\gamma)\frac{1}{q}\sum_{x\in\mathbb{F}_q}H_q\left(\frac{qh_x-\gamma}{1-\gamma}\beta\right)+\frac{1}{q}\sum_{x\in\mathbb{F}_q}H_q(qh_x\beta)\\
&\quad=\frac{1}{q}\sum_{x\in\mathbb{F}_q}G(h_x)\ .
\end{aligned}$$

It is shown that $G(h)\geq 0$ for every h such that $\gamma/q\leq h\leq \gamma/q+(1-\gamma)/(q\beta)$.

$$\begin{aligned}
\frac{\mathrm{d}G}{\mathrm{d}h}&=q\beta\log_q(q-1)-q\beta\log_q(qh\beta)-qh\beta\frac{1}{qh\beta\ln q}(q\beta)\\
&\quad+q\beta\log_q(1-qh\beta)-(1-qh\beta)\frac{1}{(1-qh\beta)\ln q}(-q\beta)\\
&\quad-q\beta\log_q(q-1)+q\beta\log_q\left(\frac{qh-\gamma}{1-\gamma}\beta\right)+(qh-\gamma)\beta\frac{1-\gamma}{(qh-\gamma)\beta\ln q}\cdot\frac{q\beta}{1-\gamma}\\
&\quad-q\beta\log_q\left(1-\frac{qh-\gamma}{1-\gamma}\beta\right)+(1-\gamma-(qh-\gamma)\beta)\frac{1-\gamma}{(1-\gamma-(qh-\gamma)\beta)\ln q}\left(-\frac{q\beta}{1-\gamma}\right)\\
&=q\beta\left(-\log_q(qh\beta)-\frac{1}{\ln q}+\log_q(1-qh\beta)+\frac{1}{\ln q}\right.\\
&\qquad\left.+\log_q\left(\frac{qh-\gamma}{1-\gamma}\beta\right)+\frac{1}{\ln q}-\log_q\left(1-\frac{qh-\gamma}{1-\gamma}\beta\right)-\frac{1}{\ln q}\right)\\
&=q\beta\left(-\log_q(qh\beta)+\log_q(1-qh\beta)+\log_q\left(\frac{qh-\gamma}{1-\gamma}\beta\right)-\log_q\left(1-\frac{qh-\gamma}{1-\gamma}\beta\right)\right)\\
&=q\beta\left(\log_q\left(\frac{1}{qh\beta}-1\right)-\log_q\left(\frac{1-\gamma}{(qh-\gamma)\beta}-1\right)\right)\ .
\end{aligned}$$

It is easy to see that

$$\frac{\mathrm{d}G}{\mathrm{d}h}\begin{cases}<0 & \text{if } \gamma<qh<1\\ =0 & \text{if } qh=1\\ >0 & \text{if } 1<qh<\gamma+(1-\beta)/\gamma\ .\end{cases}$$

If $qh=1$, then $G(1/q)=0$. This completes the proof.