# Generalization of BJMM-ISD Using May-Ozerov Nearest Neighbor Algorithm over an Arbitrary Finite Field $\mathbb{F}_q$

Cheikh Thiécoumba Gueye[1], Jean Belo Klamti[1(✉)], and Shoichi Hirose[2]

[1] Faculté des Sciences et Techniques, DMI, LACGAA,
Université Cheikh Anta Diop, Dakar, Senegal
{cheikht.gueye,jeanbelo.klamti}@ucad.edu.sn
[2] Graduate School of Engineering, University of Fukui, Fukui, Japan
hrs_shch@u-fukui.ac.jp

**Abstract.** The security of McEliece cryptosystem heavily relies on the hardness of decoding a random linear code. The best known generic decoding algorithms are derived from the Information-Set Decoding (ISD) algorithm. The ISD algorithm was proposed in 1962 by *Prange* and improved in 1989 by *Stern* and later in 1991 by *Dumer*. Since then, there have been numerous works improving and generalizing the ISD algorithm: *Peters* in 2009, *May, Meurer* and *Thomae* in 2011, *Becker, Joux, May* and *Meurer* in 2012, *May* and *Ozerov* in 2015, and *Hirose* in 2016. Among all these improvement and generalization only those of*Peters* and *Hirose* are over $\mathbb{F}_q$ with $q$ an arbitrary prime power. In *Hirose*'s paper, he describes the *May-Ozerov* nearest-neighbor algorithm generalized to work for vectors over the finite field $\mathbb{F}_q$ with arbitrary prime power $q$. He also applies the generalized algorithm to the decoding problem of random linear codes over $\mathbb{F}_q$. And he observed by a numerical analysis of asymptotic time complexity that the *May-Ozerov* nearest-neighbor algorithm may not contribute to the performance improvement of Stern's ISD algorithm over $\mathbb{F}_q$ with $q \geq 3$. In this paper, we will extend the *Becker, Joux, May*, and *Meurer*'s ISD using the *May-Ozerov* algorithm for Nearest-Neighbor problem over $\mathbb{F}_q$ with $q$ an arbitrary prime power. We analyze the impact of May-Ozerov algorithm for Nearest-Neighbor Problem over $\mathbb{F}_q$ on *the Becker, Joux, May* and *Meurer*'s ISD.

**Keywords:** Code-based cryptography · Information-Set Decoding (ISD) algorithm · Linear code · Nearest neighbor

## 1 Introduction

Code-based cryptography introduced by McEliece [29] is one of the most promising solution for designing secure cryptosystems against quantum attacks. The McEliece public-key encryption scheme, based on binary Goppa codes, has so far successfully resisted all cryptanalysis efforts. But it is not used in real life because


S. El Hajji et al. (Eds.): C2SI 2017, LNCS 10194, pp. 96–109, 2017.
DOI: 10.1007/978-3-319-55589-8_7

of the key length problem. In order to decrease the public-key size, some variants were proposed by concentrating on subclasses of alternant/Goppa codes which admit very compact public matrices, typically quasi-cyclic (QC), quasi-dyadic (QD), or quasi-monoidic (QM) matrices [2,14,18,27,28,30,36]. The security of the McEliece cryptosystem relies on the fact that the public key does not have any known structure. The attacker is faced with the problem of decoding a random code. A way to do this decoding is to use the Information-Set Decoding (ISD) algorithm. The ISD algorithm was introduced by Prange in 1962 [38]. Its principle is to find an information set where there are no errors positions. Its target is to answer to the Computational Syndrome Decoding (CSD) Problem.

In this paper, we will extend the best version of the ISD attack algorithm to arbitrary code over $\mathbb{F}_q$ and analyze the security of such codes to this new improved version. It is important to note that Peters used the ISD attack to prove the security of arbitrary codes over $\mathbb{F}_q$ [37], later *Ayoub et al.* introduced a polynomial attack against Wild McEliece over quadratic extensions and their attack is a structural attack [9]. Recently, *Hirose* applied the *May-Ozerov* algorithm for Nearest-Neighbor problem over $\mathbb{F}_q$ to generalize Stern's ISD version and he observed that the *May-Ozerov* algorithm for Nearest-Neighbor problem may not contribute to improve Stern's ISD [19]. The contribution of our paper is the generalization of Becker, Joux, May, and Meurer's ISD using the May-Ozerov algorithm for Nearest-Neighbor problem [32] over $\mathbb{F}_q$ with $q$ an arbitrary prime power. We analyze the contribution of the *May-Ozerov* algorithm for Nearest-Neighbor problem over an arbitrary finite field $\mathbb{F}_q$ to the performance of *Becker, Joux, May, and Meurer*'s ISD. And we analyze the security over an arbitrary finite field $\mathbb{F}_q$.

**$q$-ary Computational Syndrome Decoding (CSD) Problem**
**Input:** $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\boldsymbol{s} \in \mathbb{F}_q^{n-k}$ and an integer $\omega > 0$.
**Output:** Find $\boldsymbol{e} \in \mathbb{F}_q^n$ of weight $\leq \omega$ such that $\mathbf{H}\boldsymbol{e}^T = \boldsymbol{s}$

**Information-Set Decoding (ISD) Algorithm.** The best known attacks against the classical McEliece code-based cryptosystem are generic decoding attacks that treat McEliece's hidden binary Goppa codes as random linear codes. Introduced by Prange in 1962 (see [38]), the ISD algorithm is a generic decoding attack algorithm. Its target is to solve the CSD problem taking only as inputs a basis of the code and a noisy codeword. Improvements of this form of ISD were developed by Lee and Brickell [25], Stern [40], May, Meurer and Thomae [31], Becker, Joux, May and Meurer (BJMM-ISD) [4], later by May and Ozerov [32] used the nearest neighbor algorithm to improve the BJMM-ISD.

**Organisation of Paper.** The paper is organized as follows: in Sect. 2, we give some definitions and notations on coding theory, in Sect. 3 we give a summary of previous and recent results on ISD algorithm over an arbitrary finite fields $\mathbb{F}_q$. In Sect. 4, we give the version of BJMM-ISD using the May-Ozerov Nearest Neighbor algorithm. And in Sect. 5, we give the asymptotic complexity of our algorithm.

## 2 Coding Theory Background

### 2.1 Definitions and Notations

Let $\mathbb{F}_q$ be a finite field ($q = p^m$, $p$ is prime). A $q$-ary linear code $\mathcal{C}$ of length $n$ and dimension $k$ over $\mathbb{F}_q$ is a vectorial subspace of dimension $k$ of the full vectorial space $\mathbb{F}_q^n$. It can be specified by a full rank matrix $\mathbf{G} \in \mathbb{F}_q^{k\times n}$ called generator matrix of $\mathcal{C}$ whose rows span the code. Namely, $C = \left\{\boldsymbol{x}\mathbf{G} \;\; such \;\; that \;\; \boldsymbol{x} \in \mathbb{F}_q^k\right\}$.

A linear code can be also defined by the right kernel of matrix $\mathbf{H}$ called parity-check matrix of $\mathcal{C}$ as follows:

$$\mathcal{C} = \left\{\boldsymbol{x} \in \mathbb{F}_q^n \;\; such \;\; that \;\; \mathbf{H}\boldsymbol{x}^T = \boldsymbol{0}\right\}.$$

The *Hamming distance* between two codewords is the number of positions (coordinates) where they differ. The minimal distance of a code is the minimal distance of all codewords. The *weight* of a word $\boldsymbol{x} \in \mathbb{F}_q^n$ denote by $wt\,(\boldsymbol{x})$ is the number of its nonzero positions. Then the minimal *weight* of a code $\mathcal{C}$ is the minimal *weight* of all codewords. If a code $\mathcal{C}$ is linear, the minimal distance is equal to the minimal *weight* of the code.

Let $\mathcal{C}$ be a $q$-ary linear code of length $n$, dimension $k$ and generator matrix $\mathbf{G} = \left(\boldsymbol{g}_0, \boldsymbol{g}_1, ..., \boldsymbol{g}_{n-1}\right)$ with $\boldsymbol{g}_i \in \mathbb{F}_q^n$ for all $i \in \{0, 1, \ldots, n-1\}$. Let $I \subset \{0, 1, \ldots, n-1\}$ with $|I| = k$. We call $I$ an *information set* if and only if the matrix $\mathbf{G}_I = (\boldsymbol{g})_{i\in I}$ is inversible.

A vector $\boldsymbol{u} \in \mathbb{F}_q^\ell$ is called a balanced vector if the number of its coordinates equal to $x$ is $\ell/q$ for all $x \in \mathbb{F}_q$.

For $\boldsymbol{x} = (x_1, ..., x_n) \in \mathbb{F}_q^n$ and a non zero integer $j < n$, let $x_{[j]} = (x_1, ..., x_j)$ and $\boldsymbol{x}^{[j]} = (x_{n-j+1}, ..., x_n)$.

We denote the $q$-ary entropy function by:

$$H_q(x) = x\log(q-1) - x\log(x) - (1-x)\log(1-x)$$

For all integer $n$, let $[n] = \{1, \ldots, n\}$. If $I$ is a subset of $[n]$, for all vector $\boldsymbol{x} = (x_1, ..., x_n)$, let $\boldsymbol{x}_I = (x_i)_{i\in I}$.

### 2.2 McEliece's Cryptosystem

McEliece's cryptosystem is a public-key encryption scheme introduced in 1978 by *McEliece*. The original version used the Goppa binary code remained unbroken. It can also be used with any class of codes which has an efficient decoding algorithm.

**Secret keys:** A matrix $\mathbf{G} \in \mathbb{F}_2^{k\times n}$, $\mathbf{S} \in \mathbb{F}_2^{k\times k}$ (an invertible matrix), $\mathbf{P} \in \mathbb{F}_2^{n\times n}$ (a random permutation matrix).
**Public keys:** The matrix $\tilde{\mathbf{G}} = \mathbf{SGP}$ and the corrector capacity $t$.
**Encryption:** Let $\boldsymbol{m}$ be a plaintext then the ciphertext $\boldsymbol{c}$ is given by:

$$\boldsymbol{c} = \boldsymbol{m}\tilde{\mathbf{G}} + \boldsymbol{e}$$

with $\boldsymbol{e}$ a $q$-ary vector of length $n$ and weight $t$.

**Decryption:** Compute

$$\tilde{\boldsymbol{c}} = \boldsymbol{m}\tilde{\mathbf{G}}\mathbf{P}^{-1} + \boldsymbol{e}\mathbf{P}^{-1}$$

and use the decoding algorithm to find $\tilde{\boldsymbol{m}} = \boldsymbol{m}\mathbf{S}$ and finally find $\boldsymbol{m}$ by computing $\boldsymbol{m} = \tilde{\boldsymbol{m}}\mathbf{S}^{-1}$.

### 2.3 Nearest-Neighbor Problem

The nearest-neighbor (NN) problem over the binary field defined in [32] is generalized over other finite fields in [19].

*Neartest Neighbor Problem over* $\mathbb{F}_q$*:* Let $q$ be a prime power. Let $m$ be a positive integer. Let $0 < \gamma < 1/2$ and $0 < \lambda < 1$. Then $(m, \gamma, \lambda)$-NN problem is defined by:

**Input:** The constant $\gamma$ and two lists $U \subset \mathbb{F}_q^m$, $V \subset \mathbb{F}_q^m$ of size $|U| = |U| = q^{\lambda n}$ with uniform and pairwise independent vectors.
**Output:** $\mathcal{C} \subset U \times V$ which has $(\boldsymbol{u}, \boldsymbol{v})$ such that $wt(\boldsymbol{u} - \boldsymbol{v}) = \gamma m$ with $wt\,(\boldsymbol{u} - \boldsymbol{v})$ is the weight of $\boldsymbol{u} - \boldsymbol{v}$.

## 3 Preview Work on Information-Set Decoding over $\mathbb{F}_q$

We denote in the rest of the paper the concatenation of two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ (respectively of two matrices $\mathbf{A}$ and $\mathbf{B}$) by $(\boldsymbol{x}|\boldsymbol{y})$ (respectively $(\mathbf{A}|\mathbf{B})$).

In this section we give a survey of the generalization of ISD algorithm over an arbitrary finite field.

***Peters:*** In 2009, *Peters* was the first to propose a generalization of the ISD algorithm over an arbitrary finite field $\mathbb{F}_q$. In her paper [37], she proposed the generalization of Stern-ISD which all of the ISD improvements are based on.

***Cayrel et al.:*** In 2010 just few months after *Peters*'s paper, *Cayrel et al.* [34] improved the performance of the ISD over an arbitrary finite field by giving a lower bound of ISD algorithm and they generalized the formula of the lower bound introduced by *Finiasz et al.* in [15].

***Meurer:*** In 2012 just after their ISD algorithm in the binary case in [4,31], *Meurer* proposed a new generalization of the ISD algorithm over an arbitrary finite field in his dissertation thesis [33] based on these two papers.

***Hirose:*** In 2016 *Hirose* gave a generalization of the *nearest-neighbor* algorithm introduced by *May-Ozerov* [32] to generalize the Stern-ISD algorithm. And he analyzed the contribution of the *May-Ozerov*'s nearest-neighbor algorithm over an arbitrary finite field to the performance of Stern-ISD algorithm over an arbitrary finite field.

The following tables give us a summary complexity results on the ISD algorithm generalization previous work. We denote the ISD algorithm generalization given by*Peters* by $q$-Stern-ISD, *Hirose*'s generalization by $q$-Hirose-ISD.

**Table 1.** Complexity of ISD algorithm over an arbitrary finite field given in [19].

| $q$ | $q$-Stern-ISD | $q$-Hirose-ISD |
|---|---|---|
| | Half distance | Half distance |
| 2 | 0.05563 | 0.05498 |
| 3 | 0.05217 | 0.05242 |
| 4 | 0.04987 | 0.05032 |
| 5 | 0.04815 | 0.04864 |
| 7 | 0.04571 | 0.04614 |
| 8 | 0.04478 | 0.04519 |
| 8 | 0.04266 | 0.04299 |

**Table 2.** Complexity of ISD algorithm over an arbitrary finite field given by Meurer in [33].

| $q$ | $q$-Meurer-ISD | |
|---|---|---|
| | BReps | XBReps |
| 2 | 0.1053 | - |
| 4 | 0.1033 | 0.1014 |
| 8 | 0.0989 | 0.0969 |
| 16 | 0.0929 | 0.0918 |
| 32 | 0.0867 | 0.0863 |
| 64 | 0.0808 | 0.0806 |

In *Meurer*'s dissertation thesis, he gave two variants of ISD algorithm generalization then we denote the basic variant by BReps and the extended variant by XBReps.

## 4 Becker, Joux, May and Meurer ISD Using May-Ozerov Nearest-Neighbor Algorithm over $\mathbb{F}_q$

The Becker, Joux, May and Meurer ISD using May-Ozerov Nearest-Neighbor algorithm over an arbitrary finite field $\mathbb{F}_q$ is presented in *Algorithm* 1.

In this algorithm we construct Base Lists over $\mathbb{F}_q$ like in [4]. For all $j = 0,\ 1$ we denote the Base Lists by $\mathcal{B}_{j,1}^{\mathcal{L}_j}$, $\mathcal{B}_{j,2}^{\mathcal{L}_j}$, $\mathcal{B}_{j,1}^{\mathcal{R}_j}$ and $\mathcal{B}_{j,2}^{\mathcal{R}_j}$. We define $\mathcal{B}_{j,1}^{\mathcal{L}_j}$ as follows:

Let $\mathcal{P}_{j,1}^{\mathcal{L}_j}$ and $\mathcal{P}_{j,2}^{\mathcal{L}_j}$ be be a partition of $[k+\ell] = \{1, ..., k+\ell\}$ such that $\left|\mathcal{P}_{j,1}^{\mathcal{L}_j}\right| = \left|\mathcal{P}_{j,2}^{\mathcal{L}_j}\right| = \dfrac{k+\ell}{2}$ then

$$\mathcal{B}_{j,1}^{\mathcal{L}_j} = \left\{ \boldsymbol{x} \in \mathbb{F}_q^{k+\ell} \times \left\{ \boldsymbol{0}^{n-k-\ell} \right\} \ s.t \ \ wt\left(\boldsymbol{x}\right) = \frac{p}{8} + \frac{\epsilon_1}{4} + \frac{\epsilon_2}{2} \ \ with \ \ \boldsymbol{x}_{P_{j,2}^{\mathcal{L}_j}} = (0, 0, ..., 0) \right\}$$

Where $p$, $\epsilon_1$ and $\epsilon_2$ are the parameters of the algorithm such that $0 \leq p < k+\ell$, $0 < \epsilon_1 < k+\ell-p$, $0 < \epsilon_2 < k+\ell-\frac{p}{2}-\epsilon_1$. The construction of $\mathcal{B}_{j,2}^{\mathcal{L}_j}$, $\mathcal{B}_{j,1}^{\mathcal{R}_j}$ and $\mathcal{B}_{j,2}^{\mathcal{R}_j}$ is similar.

We use these Base Lists to compute a vector $\boldsymbol{e} \in \mathbb{F}_q^{k+\ell} \times \left\{\boldsymbol{0}^{n-k-\ell}\right\}$ such that $wt\left(\boldsymbol{e}_{[k+\ell]}\right) = p$ and $\boldsymbol{e} = \boldsymbol{e}_1 - \boldsymbol{e}_2$ with $\boldsymbol{e}_1$, $\boldsymbol{e}_2 \in \mathbb{F}_q^{k+\ell} \times \left\{\boldsymbol{0}^{n-k-\ell}\right\}$ and $wt\left(\boldsymbol{e}_1\right) = wt(\boldsymbol{e}_2) = \frac{p}{2} + \epsilon_1$.

**Proposition 1** *[33]. Let $0 \leq p \leq k+\ell$ be an integer and $\boldsymbol{e} \in \mathbb{F}_q^{k+\ell} \times \left\{\boldsymbol{0}^{n-k-\ell}\right\}$ be a vector such that $wt\left(\boldsymbol{e}\right) = p$. For all integer $\epsilon$ such that $0 \leq \epsilon < k+\ell-p$, denote $\vartheta\left(k,\ell,\epsilon,p,q\right)$ the number of pairs $\left(\boldsymbol{e}_1,\boldsymbol{e}_2\right)$ such that $\boldsymbol{e} = \boldsymbol{e}_1 - \boldsymbol{e}_2$ with $\boldsymbol{e}_1$, $\boldsymbol{e}_2 \in \mathbb{F}_q^{k+\ell} \times \left\{\boldsymbol{0}^{n-k-\ell}\right\}$ and $wt\left(\boldsymbol{e}_1\right) = wt(\boldsymbol{e}_2) = \frac{p}{2} + \epsilon$. It holds*

$$\vartheta\left(k,\ell,\epsilon,p,q\right) = \sum_{i=0}^{\min\left(\frac{p}{2},\epsilon\right)} \binom{p-2i}{\frac{p}{2}-i} (q-2)^{2i} \binom{k+\ell-p}{\epsilon-i} (q-1)^{\epsilon-i}$$

Then $\vartheta\left(k,\ell,\epsilon,p,q\right) \geq \binom{p}{\frac{p}{2}}\binom{k+\ell-p}{\epsilon}(q-1)^{\epsilon}$.

And asymtopticalLy by using the inequality $\log_q 2 < H_q\left(\frac{1}{2}\right)$, we implicity lower bound $\log_q \vartheta\left(k,\ell,\epsilon,p,q\right) \geq p\log_q 2 + (k+\ell-p)\, H_q\left(\frac{\epsilon}{k+\ell-p}\right)$ [33]. This brief analysis will allow us to give a constraint on some parameters of our algorithm.

---

**Algorithm 1.** $q$-BJMM-MO algorithm over $\mathbb{F}_q$

---

**Constants:** Let $n$, $k$, $d$ and $\omega$ be nonzero integers such that $k \leq n$ and $\omega = \lfloor \frac{d-1}{2} \rfloor$ with $d = H_q^{-1}\left(1 - \frac{k}{n}\right)$

**Parameters:** Integers $p$, $\ell$, $r_1$, $\epsilon_1$ and $\epsilon_2$ such that $0 \leq p \leq \min\left\{k+\ell,\omega\right\}$, $0 < r_1 < \ell \leq \min\left\{n-k-\omega+p, n-k\right\}$, $0 < \epsilon_1 < k+\ell-p$ and $0 < \epsilon_2 < k+\ell-\frac{p}{2}-\epsilon_1$.

**Input:** two nonzero integers $n$ and $k$, a matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, and a nonzero vector $\boldsymbol{x} \in \mathbb{F}_q^n$.

**Output:** A vector $\boldsymbol{e} \in \mathbb{F}_q^n$ of weight $wt\left(\boldsymbol{e}\right) = \omega$ such that $\mathbf{H}\boldsymbol{e}^T = \mathbf{H}\boldsymbol{x}^T$.

1 : **Procedure:** BJMM-MO($n$, $k$, $\mathbf{H}$, $\boldsymbol{x}$)
2 : $\quad \boldsymbol{s} \longleftarrow \mathbf{H}\boldsymbol{x}^T$
3 : $\quad d \longleftarrow nH^{-1}\left(1-\frac{k}{n}\right)$
4 : $\quad \omega \longleftarrow \lfloor \frac{d-1}{2} \rfloor$
5 : $\quad$ Choose parameters $p$, $\epsilon_1$, $\epsilon_2$, $0 < r_1 < \ell < n-k$.
6 : $\quad$ **Repeat:**
7 : $\quad\quad \pi \longleftarrow$ a random permutation on $\{1,2,...,n\}$.
8 : $\quad\quad (\mathbf{Q}_1|\mathbf{Q}_2) \longleftarrow \pi\left(\mathbf{H}\right)$ with $\mathbf{Q}_2 \in \mathbb{F}_q^{(n-k)\times(n-k)}$ and $\mathbf{Q}_1 \in \mathbb{F}_q^{(n-k)\times k}$
9 : $\quad\quad$ **While** $\mathbf{Q}_2$ is not invertible:

$\pi \longleftarrow$ a random permutation on $\{1, 2, ..., n\}$.
$(\mathbf{Q}_1|\mathbf{Q}_2) \longleftarrow \pi(\mathbf{H})$
$\tilde{\mathbf{H}} \longleftarrow \mathbf{Q}_2^{-1}\pi(\mathbf{H})$ and $\tilde{\boldsymbol{s}} \longleftarrow \mathbf{Q}_2^{-1}\boldsymbol{s}$
Choose randomly $\boldsymbol{t}_{\mathcal{L}} \in \mathbb{F}_2^{\ell}$ and $\boldsymbol{t}_{\mathcal{L}_0}, \boldsymbol{t}_{\mathcal{R}_0} \in \mathbb{F}_q^{r_1}$
Compute $\boldsymbol{t}_{\mathcal{R}} = \boldsymbol{t}_{\mathcal{L}} - \tilde{\boldsymbol{s}}_{[\ell]}$, $\boldsymbol{t}_{\mathcal{L}_1} = \boldsymbol{t}_{\mathcal{L}_0} - (\boldsymbol{t}_{\mathcal{L}})_{[r_1]}$ and $\boldsymbol{t}_{\mathcal{R}_1} = \boldsymbol{t}_{\mathcal{R}_0} - (\boldsymbol{t}_{\mathcal{R}})_{[r_1]}$.
Compute Base Lists $\mathcal{B}_{i,1}^{\mathcal{L}_i}$, $\mathcal{B}_{i,2}^{\mathcal{L}_i}$, $\mathcal{B}_{i,1}^{\mathcal{R}_i}$and $\mathcal{B}_{i,1}^{\mathcal{R}_i}$ , with $i = 0, 1$ and:
$\mathcal{L}_i \longleftarrow \left\{\boldsymbol{u} = \boldsymbol{a} - \boldsymbol{b} \;\; s.t \;\; \boldsymbol{a} \in \mathcal{B}_{i,1}^{\mathcal{L}_i},\ \boldsymbol{b} \in \mathcal{B}_{i,2}^{\mathcal{L}_i} \;\; with \;\; wt(\boldsymbol{u}) = \frac{p}{4} + \frac{\epsilon_1}{2} + \epsilon_2, \right.$
$\left. and \;\; \left(\tilde{\mathbf{H}}\boldsymbol{u}^T\right)_{[r_1]} = \boldsymbol{t}_{\mathcal{L}_i}\right\}$
$\mathcal{R}_i \longleftarrow \left\{\boldsymbol{u} = \boldsymbol{a} - \boldsymbol{b} \;\; s.t \;\; \boldsymbol{a} \in \mathcal{B}_{i,1}^{\mathcal{R}_i},\ \boldsymbol{b} \in \mathcal{B}_{i,2}^{\mathcal{R}_i} \;\; with \;\; wt(\boldsymbol{u}) = \frac{p}{4} + \frac{\epsilon_1}{2} + \epsilon_2, \right.$
$\left. and \;\; \left(\tilde{\mathbf{H}}\boldsymbol{u}^T\right)_{[r_1]} = \boldsymbol{t}_{\mathcal{R}_i}\right\}$
$\mathcal{L} \longleftarrow \left\{ \left(\tilde{\mathbf{H}}\boldsymbol{z}^T\right)^{[n-k-\ell]} \quad s.t \;\; \boldsymbol{z} = \boldsymbol{u} - \boldsymbol{v} \;\; and \;\; (\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{L}_0 \times \mathcal{L}_1 \right.$
$\left. \text{with } \left(\tilde{\mathbf{H}}\boldsymbol{z}^T\right)_{[\ell]} = \boldsymbol{t}_{\mathcal{L}} \right\}$
$\mathcal{R} \longleftarrow \left\{ \left(\tilde{\mathbf{H}}\boldsymbol{z}^T + \tilde{\boldsymbol{s}}\right)^{[n-k-\ell]} \quad s.t \;\; \boldsymbol{z} = \boldsymbol{u} - \boldsymbol{v} \;\; and \;\; (\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{R}_0 \times \mathcal{R}_1 \right.$
$\left. \text{with } \left(\tilde{\mathbf{H}}\boldsymbol{z}^T\right)_{[\ell]} = \boldsymbol{t}_{\mathcal{R}} \right\}$
In 18 and 19 we keep only elements with $wt(\boldsymbol{z}) = \frac{p}{2} + \epsilon_1$
$\mathcal{C} \longleftarrow$ MO-NN$\left(\mathcal{L}, \mathcal{R}, \frac{\omega - p}{n-k-\ell}\right)$
**For all** $(\boldsymbol{u}, \boldsymbol{v}) \in \mathcal{C} \cap \mathcal{L} \times \mathcal{R}$ :
Find $(\boldsymbol{e}_1, \boldsymbol{e}_2)$ s.t $\boldsymbol{u} = \left(\tilde{\mathbf{H}}\boldsymbol{e}_1^T\right)^{[n-k-\ell]}$ and $\boldsymbol{v} = \left(\tilde{\mathbf{H}}\boldsymbol{e}_2^T + \tilde{\boldsymbol{s}}\right)^{[n-k-\ell]}$
**If** $wt(\boldsymbol{e}_1 - \boldsymbol{e}_2) = p$:
**Return** $\pi^{-1}\left(\boldsymbol{e}_1 - \boldsymbol{e}_2 - \left(\boldsymbol{0}^{k+\ell}|\boldsymbol{u} - \boldsymbol{u}\right)\right)$
**End Procedure**

---

The complexity the $q$-BJMM-MO is given by:

**Theorem 1.** *Let $\varepsilon > 0$ be a real. The $q$-BJMM-MO algorithm solves the Syndrome Decoding problem of random $[n, k]$-linear code over $\mathbb{F}_q$ with overwhelming probability in time*

$$\tau(q, n, k, p, \omega, h_x, \varepsilon) = \tilde{\mathcal{O}}\left(q^{n\tau_1}\left(q^{n\tau_2} + q^{2n\tau_2 - r_1} + q^{4n\tau_2 - r_1 - \ell} + q^{n\mu} + q^{(y+\varepsilon)(n-k-\ell)}\right)\right)$$

*where*

$$\tau_1 = \left(H\left(\frac{\omega}{n}\right) - \left(\frac{k+\ell}{n}\right)H\left(\frac{p}{k+\ell}\right) - \left(1 - \frac{k+\ell}{n}\right)H\left(\frac{\omega - p}{n-k-\ell}\right)\right)\log_q 2,$$

$$\tau_2 = \frac{k+\ell}{2n}H_q\left(\frac{\frac{p}{4} + \frac{\epsilon_1}{2} + \epsilon_2}{k+\ell}\right) \quad and \quad \mu = \frac{k+\ell}{n}H_q\left(\frac{\frac{p}{2} + \epsilon_1}{k+\ell}\right) - \frac{\ell}{n}$$

*with*

$$y = (1-\gamma)\left(H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q\left(\frac{qh_x-\gamma}{1-\gamma}\beta\right)\right), \quad \gamma = \frac{\omega-p}{n-k-p}, \quad 0<\beta<1,$$

$$\max\{0, \omega+k+\ell-n\} \le p \le \min\{k+\ell, \omega\}, \quad \sum_{x\in\mathbb{F}_q} h_x = 1,$$

$$\frac{\gamma}{q} < h_x < \frac{\gamma}{q} + \frac{1-\gamma}{q\beta} \quad for \quad each \quad x \in \mathbb{F}_q,$$

$$\ell = p\log_q 2 + (k+\ell-p)H_q\left(\frac{\epsilon_1}{k+\ell-p}\right) \quad and \quad \ell \le \min\{n-k-\omega+p, n-k\}$$

$$r_1 = \left(\frac{p}{2}+\epsilon_1\right)\log_q 2 + \left(k+\ell-\frac{p}{2}-\epsilon_1\right)H_q\left(\frac{\epsilon_2}{k+\ell-\frac{p}{2}-\epsilon_1}\right)$$

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

*Proof.* Recall that

$$\mathcal{T}(q,n,k,p,\omega,h_x,\varepsilon) = \frac{1}{\mathbb{P}(\pi_{succ})}\mathcal{C}_{in}$$

where $\mathbb{P}(\pi_{succ})$ is a the probability to have the good permutation (permutation allowing to have a success decoding) and $\mathcal{C}_{in}$ is the cost of each iteration with:

$$\mathbb{P}(\pi_{succ}) = \tilde{\mathcal{O}}\left(\frac{\binom{k+\ell}{p}\binom{n-k-\ell}{\omega-p}}{\binom{n}{\omega}}\right) \Longrightarrow \frac{1}{\mathbb{P}(\pi_{succ})} = \tilde{\mathcal{O}}\left(\frac{\binom{n}{\omega}}{\binom{k+\ell}{p}\binom{n-k-\ell}{\omega-p}}\right).$$

Using the equality

$$\binom{n}{k} = 2^{nH\left(\frac{k}{n}\right)}$$

with $H$ the binary entropie function.

$$\begin{aligned}\mathbb{P}(\pi_{succ}) &= \tilde{\mathcal{O}}\left(2^{n\left(H\left(\frac{\omega}{n}\right)-\frac{k+\ell}{n}H\left(\frac{p}{k+\ell}\right)-\left(1-\frac{k+\ell}{n}\right)H\left(\frac{\omega-p}{n-k-\ell}\right)\right)}\right)\\ &= \tilde{\mathcal{O}}\left(q^{n\left(H\left(\frac{\omega}{n}\right)-\frac{k+\ell}{n}H\left(\frac{p}{k+\ell}\right)-\left(1-\frac{k+\ell}{n}\right)H\left(\frac{\omega-p}{n-k-\ell}\right)\right)\log_q 2}\right)\\ &= \tilde{\mathcal{O}}\left(q^{n\tau_1}\right).\end{aligned}$$

Let us examine the complexity of each iteration. First we construct Base Lists and the cardinality of each Base List is given by, for each $i = 1, 2$ and $j = 1, 2$

$$|\mathcal{B}_{j,i}^{\mathcal{L}_j}| = \binom{\frac{k+\ell}{2}}{\frac{p}{8}+\frac{\epsilon_1}{4}+\frac{\epsilon_2}{2}}(q-1)^{\frac{p}{8}+\frac{\epsilon_1}{4}+\frac{\epsilon_2}{2}}.$$

Then by using the equality

$$\binom{n}{k}(q-1)^k = \tilde{\mathcal{O}}\left(q^{nH_q(\frac{k}{n})}\right),$$

the complexity to compute Base Lists is given by

$$\tilde{\mathcal{O}}\left(q^{n\left(\frac{k+\ell}{2n}H_q\left(\frac{\frac{p}{4}+\frac{\epsilon_1}{2}+\epsilon_2}{k+\ell}\right)\right)}\right) = \tilde{\mathcal{O}}\left(q^{n\tau_2}\right).$$

Second we use Base Lists to make a filtering to compute $\mathcal{L}_i$ and $\mathcal{R}_i$ for each $i = 1, 2$ and the cost of this filtering is given by:

$$\tilde{\mathcal{O}}\left(\frac{|\mathcal{B}_{i,1}^{\mathcal{L}_i}||\mathcal{B}_{i,2}^{\mathcal{L}_j}|}{q^{r_1}}\right) = \tilde{\mathcal{O}}\left(q^{2n\tau_2-r_1}\right).$$

Third we compute the lists $\mathcal{L}$ and $\mathcal{R}$ with a filtering and the cost of this filtering is given by

$$\tilde{\mathcal{O}}\left(\frac{|\mathcal{L}_i||\mathcal{L}_j|}{q^{\ell-r_1}}\right) = \tilde{\mathcal{O}}\left(q^{4n\tau_2-r_1-\ell}\right).$$

Line 20 only gives the upper bound on $|\mathcal{L}| = |\mathcal{R}|$.

$$\tilde{\mathcal{O}}\left(\frac{\binom{k+\ell}{\frac{p}{2}+\epsilon_1}(q-1)^{\frac{p}{2}+\epsilon_1}}{q^\ell}\right) = \tilde{\mathcal{O}}\left(q^{n\left(\frac{k+\ell}{n}H_q\left(\frac{\frac{p}{2}+\epsilon_1}{k+\ell}\right)-\frac{\ell}{n}\right)}\right) = \tilde{\mathcal{O}}\left(q^{\mu n}\right).$$

And finally we make a last filtering using the May-Ozerov Nearest Neighbor algorithm and the cost of this filtering is given by:

$$\tilde{\mathcal{O}}\left(q^{(y+\varepsilon)(n-k-\ell)}\right).$$

We have $|\mathcal{L}| = |\mathcal{R}| = q^{\mu n}$. Thus MO-NN is given an instance of $(m, \gamma, \lambda)$-NN problem with:

$$m = n-k-\ell, \quad \gamma = \frac{\omega - p}{n-k-\ell}$$

and

$$\lambda = \frac{\mu n}{n-k-\ell}.$$

According to *Lemma* 3 in [19] we must have

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

## 5 Numerical Analysis of Time Complexity

We give in this section a optimization numerical time complexity of our algorithm in the half distance decoding using the code's parameters given in [19] and in the full distance decoding using the code's parameters given in [33]. We give these complexities for $q \geq 3$ because the case $q = 2$ is already done in [4,31–33]

**Table 3.** Complexity of the $q$-BJMM-MO algorithm in the half distance decoding for parameters in [19].

| $q$ | $q$-BJMM-MO | | | | | |
|---|---|---|---|---|---|---|
| | $c_k$ | $c_\ell$ | $c_p$ | $h$ | $\beta$ | Half dist. |
| 3 | 0.4545 | 0.06273 | 0.015678 | 0.104457 | 0.081899 | 0.04427 |
| 4 | 0.4625 | 0.05936 | 0.012787 | 0.109280 | 0.065891 | 0.04194 |
| 5 | 0.4727 | 0.05664 | 0.010710 | 0.119404 | 0.059101 | 0.03955 |
| 7 | 0.4812 | 0.05383 | 0.009768 | 0.103261 | 0.042989 | 0.03706 |
| 8 | 0.4891 | 0.05232 | 0.008728 | 0.116760 | 0.039019 | 0.03593 |
| 11 | 0.4959 | 0.05045 | 0.009829 | 0.093971 | 0.029929 | 0.03335 |

**Table 4.** Complexity of the $q$-BJMM-MO algorithm in the full distance decoding for parameters in [33].

| $q$ | $q$-BJMM-MO | | | | | |
|---|---|---|---|---|---|---|
| | $c_k$ | $c_\ell$ | $c_p$ | $h$ | $\beta$ | Full dist. |
| 4 | 0.4259 | 0.047749 | 0.015721 | 0.113254 | 0.058929 | 0.09951 |
| 8 | 0.4529 | 0.036823 | 0.009021 | 0.123717 | 0.019890 | 0.09388 |
| 16 | 0.4729 | 0.029908 | 0.008021 | 0.049354 | 0.021199 | 0.09012 |
| 32 | 0.4829 | 0.025151 | 0.007521 | 0.031235 | 0.014109 | 0.08264 |
| 64 | 0.4929 | 0.021496 | 0.006521 | 0.012637 | 0.013109 | 0.07861 |

## 6 Conclusion

The *May-Ozerov*'s Nearest Neighbor algoritm allows us to improve the generalization of BJMM-ISD. We show in the Tables 1 and 3 that our generalization is faster than *Hirose*'s generalization in the half distance decoding and in addition by comparing the Tables 2 and 4 we show that is faster than *Meurer*'s generalization.

**Acknowlegment.** This work was carried out with financial support of CEA-MITIC for CBC project and financial support of the government of Senegal's Ministry of Hight Education and Research for ISPQ project. The third author was supported in part by JSPS KAKENHI Grant Number JP16H02828.

## Appendix

## Nearest-Neighbor Algorithm over an Arbitrary Finite Field $\mathbb{F}_q$

We give in this section the May-Ozerov Nearest-Neighbor algorithm over $\mathbb{F}_q$ proposed by *Hirose* in [19]

**Algorithm 2.** May-Ozerov Nearest-Neighbor algorithm over $\mathbb{F}_q$

**Procedure:**MO-NN($\mathcal{L}$, $\mathcal{R}$, $\gamma$ )
  $y \longleftarrow (1-\gamma)\left(H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q\left(\frac{qh_x-\gamma}{1-\gamma}\beta\right)\right)$
  Choose $\epsilon > 0$
  $t \longleftarrow \lceil \frac{\log_2(y-\lambda+\frac{\epsilon}{2})-\log_2(\frac{\epsilon}{2})}{\log_2(y)-\log_2(\lambda)} \rceil$
  $\alpha_1 \longleftarrow \frac{y-\lambda+\frac{\epsilon}{2}}{y}$
  $\alpha_j \longleftarrow \frac{\lambda}{y}\alpha_{j-1}$ **for** $2 \le j \le t$
  **For** $m^{\mathcal{O}(1)}$ times:
    Choose a permutation $\pi$ on $\mathbb{F}_q^m$ uniformly at random
    Choose a vector $\mathbf{r} = (\mathbf{r}_1, \ldots, \mathbf{r}_t) \in \mathbb{F}_q^{\alpha_1 m} \times \ldots \times \mathbb{F}_q^{\alpha_t m} = \mathbb{F}_q^m$ uniformly at random s.t $\mathbf{r}_i$ is balanced for all $1 \le i \le t$
    $\tilde{\mathcal{L}} \longleftarrow \{\tilde{\mathbf{u}} = (\tilde{\mathbf{u}}_1, \ldots, \tilde{\mathbf{u}}_t)$ s.t $\tilde{\mathbf{u}} = \pi(\mathbf{u}) + \mathbf{r}$ with $\mathbf{u} \in \mathcal{L}$ and $\tilde{\mathbf{u}}_j$ is balanced for every $1 \le j \le t\}$
    $\tilde{\mathcal{R}} \longleftarrow \{\tilde{\mathbf{v}} = (\tilde{\mathbf{v}}_1, \ldots, \tilde{\mathbf{v}}_t)$ s.t $\tilde{\mathbf{v}} = \pi(\mathbf{v}) + \mathbf{r}$ with $\mathbf{u} \in \mathcal{R}$ and $\tilde{\mathbf{v}}_j$ is balanced for every $1 \le j \le t\}$
    **Return** MO-NNR($\tilde{\mathcal{L}}$, $\tilde{\mathcal{R}}$, $m$, $t$, $\gamma$, $\lambda$, $\alpha_1$,..., $\alpha_t$,$y$, $\epsilon$, 1 )
**End Procedure**

The complexity of May-Ozerov Nearest Neighbor algorithm is given by:

**Theorem 2** *[19]. Let $q$ be a prime power. Let $\gamma$, $\beta$, $\epsilon > 0$ and $\lambda$ be reals such that $0 < \gamma < \frac{1}{2}$, $0 < \beta < 1$, $\varepsilon > 0$ and $\lambda \le H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q(q\beta h_x)$ with $\sum_{x\in\mathbb{F}_q} h_x = 1$ and for each $x \in \mathbb{F}_q$, $\frac{\gamma}{q} < h_x < \frac{\gamma}{q} + \frac{1-\gamma}{q\beta}$.*

*Let $y = (1-\gamma)\left(H_q(\beta) - \frac{1}{q}\sum_{x\in\mathbb{F}_q} H_q\left(\frac{qh_x-\gamma}{1-\gamma}\beta\right)\right)$. Then the MO-NN algorithm solves the $(m, \gamma, \lambda)NN$ problem over $\mathbb{F}_q$ with overwhelming probability in time*

$$\tilde{\mathcal{O}}\left(q^{(y+\varepsilon)m}\right).$$

**Algorithm 3.** May-Ozerov NearestNeighborRec algorithm over $\mathbb{F}_q$

**Procedure:**MO-NNR($\mathcal{L}$, $\mathcal{R}$, $m$, $t$, $\gamma$, $\lambda$, $\alpha_1$,..., $\alpha_t$, $y$, $\epsilon$, $i$ )
  **If** $i = t + 1$:
    $\mathcal{C} \longleftarrow \{(\mathbf{u}, \mathbf{v}) \in \mathcal{L} \times \mathcal{R}$ *s.t* $wt(\mathbf{u} - \mathbf{v}) = \gamma m\}$
  **For** $\mathcal{O}(q^{y\alpha_i m})$ times:
    Choose $A_i \subset \{(\alpha_1 + \cdots + \alpha_{i-1})m + 1, \ldots, (\alpha_1 + \cdots + \alpha_i)m\}$ uniformly at random s.t $|A_i| = \beta\alpha_i m$ with

$(\alpha_1 + \cdots + \alpha_{i-1})m = 0$ if $i = 1$

$\mathcal{L}' \longleftarrow \{\mathbf{u} \in \mathcal{L}$ s.t the number of each $x \in \mathbb{F}_q$ on $A_i$ *is* $h_x \beta \alpha_i m\}$

$\mathcal{R}' \longleftarrow \{\mathbf{v} \in \mathcal{L}$ s.t the number of every $x \in \mathbb{F}_q$ on $A_i$ is $h_x \beta \alpha_i m\}$

**If** $|\mathcal{L}'| = |\mathcal{R}'| = \tilde{\mathcal{O}}\left(q^{\left(\lambda\left(1-\sum_{j=1}^{i}\alpha_j\right)+\frac{\epsilon}{2}\right)m}\right)$:

$\mathcal{C} \longleftarrow \mathcal{C} \cup \text{MO-NNR}(\mathcal{L}', \mathcal{R}', m, t, \gamma, \lambda, \alpha_1, \ldots, \alpha_t, y, \epsilon, i+1)$

**Return** $\mathcal{C}$

**End Procedure**

## References


1. Andoni, A., Indyk, P., Nguyen, H.L., Razenshteyn, I.: Beyond locality-sensitive hashing. In: SODA, pp. 1018–1028 (2014)
2. Berger, T.P., Cayrel, P.-L., Gaborit, P., Otmani, A.: Reducing key length of the McEliece cryptosystem. In: Preneel, B. (ed.) AFRICACRYPT 2009. LNCS, vol. 5580, pp. 77–97. Springer, Heidelberg (2009). doi:10.1007/978-3-642-02384-2_6
3. Berlekamp, E., McEliece, R., van Tilborg, H.: On the inherent intractability of certain coding problems. IEEE Trans. Inf. Theor. **24**(3), 384–386 (1978)
4. Becker, A., Joux, A., May, A., Meurer A.: Decoding random binary linear codes in $2n, 20$: how $1 + 1 = 0$ improves information set decoding. In: Eurocrypt 2012 (2012)
5. Bernstein, D.J., Lange, T., Peters, C.: Smaller decoding exponents: ball-collision decoding. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 743–760. Springer, Heidelberg (2011). doi:10.1007/978-3-642-22792-9_42
6. Chabot, C., Legeay, M.: Using permutation group for decoding. In: Proceedings of Algebraic and Combinatorial Coding Theory 2010, pp. 86–92 (2010)
7. Coffey, J.T., Goodman, R.M.: The complexity of Information-Set Decoding (ISD). IEEE Trans. Inf. Theor. **36**(5), 1031–1037 (1990)
8. Cohen, G., Wolfmann, J. (eds.): Coding Theory and Applications. LNCS, vol. 388. Springer, Heidelberg (1989)
9. Couvreur, A., Otmani, A., Tillich, J.-P.: Polynomial time attack on wild McEliece over quadratic extensions. Cryptology ePrint Archive 2014/112 (2014)
10. Dubiner, M.: Bucketing coding and information theory for the statistical high-dimensional nearest-neighbor problem. IEEE Trans. Inf. Theor. **56**(8), 4166–4179 (2010)
11. Dumer, I.: On minimum distance decoding of linear codes. In: Proceedings 5th Joint Soviet-Swedish International Workshop Information Theory, Moscow, pp. 50–52 (1991)
12. Faugère, J.-C., Otmani, A., Perret, L., Tillich, J.-P.: Algebraic cryptanalysis of McEliece variants with compact keys. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 279–298. Springer, Heidelberg (2010)
13. Faugére, J.-C., Otmani, A., Perret, L., de Portzamparc, F., Tillich, J.-P.: Structural cryptanalysis of McEliece schemes with compact keys. Cryptology ePrint Archive: Report 2014/210 (2014)

14. Faugére, J.C., Otmani, A., Perret, L., de Portzamparc, F., Tillich, J.P.: Folding alternant and Goppa codes with non-nrivial automorphism groups. arXiv:1405.5101v1 [cs.IT], 20 May 2014
15. Finiasz, M., Sendrier, N.: Security bounds for the design of code-based cryptosystems. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 88–105. Springer, Heidelberg (2009)
16. Johansson, T., Löndahl, C.: An Improvement to Stern's Algorithm
17. Heyse, S.: Implementation of McEliece based on quasi-dyadic goppa codes for embedded devices. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 143–162. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25405-5_10
18. Gaborit, P.: Shorter keys for code based cryptography. In: Proceedings of the 2005 International Workshop on Coding and Cryptography (WCC 2005), Bergen, Norway, pp. 81–91, March 2005
19. Hirose, S.: May-Ozerov algorithm for nearest-neighbor problem over $\mathbb{F}_q$ and its application to information set decoding. Cryptology ePrint Archive: Report 2016/237 (2016)
20. Har-Peled, S., Indyk, P., Motwani, R.: Approximate nearest neighbor: towards removing the curse of dimensionality. Theor. Comput. **8**(1), 321–350 (2012)
21. Howgrave-Graham, N., Joux, A.: New generic algorithms for hard knapsacks. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 235–256. Springer, Heidelberg (2010)
22. Kobara, K.: Flexible quasi-dyadic code-based public-key encryption and signature. Cryptology ePrint Archive, Report 2009/635 (2009)
23. Legeay, M.: Permutation decoding: towards an approach using algebraic properties of the $\sigma$-subcode. In: Augot, D., Canteaut, A. (eds.) WCC 2011, pp. 193–202 (2011)
24. Legeay, M.: Utilisation du groupe de permutations d'un code correcteur pour améliorer l'éfficacité du décodage. Université de Rennes 1, Année (2012)
25. Lee, P.J., Brickell, E.F.: An observation on the security of McEliece's public-key cryptosystem. In: Barstow, D., et al. (eds.) EUROCRYPT 1988. LNCS, vol. 330, pp. 275–280. Springer, Heidelberg (1988). doi:10.1007/3-540-45961-8_25
26. Leon, J.S.: A probabilistic algorithm for computing minimum weights of large error-correcting codes. IEEE Trans. Inf. Theor. **34**, 1354–1359 (1988)
27. Misoczki, R., Barreto, P.S.L.M.: Compact McEliece keys from Goppa codes. In: Jacobson, M.J., Rijmen, V., Safavi-Naini, R. (eds.) SAC 2009. LNCS, vol. 5867, pp. 376–392. Springer, Heidelberg (2009)
28. Misoczki, R., Tillich, J.P, Sendrier, N., Barreto, P.S.L.M.: MDPC-McEliece: new McEliece variants from moderate density parity-check codes. In: ISIT 2013, pp. 2069–2073 (2013)
29. McEliece, R.: A public-key cryptosystem based on algebraic coding theory. DSN Prog. Rep., Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, pp. 114–116, January 1978
30. Barreto, P.S.L.M., Lindner, R., Misoczki, R.: Monoidic codes in cryptography. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 179–199. Springer, Heidelberg (2011)
31. May, A., Meurer, A., Thomae, E.: Decoding random linear codes in $\tilde{\mathcal{O}}(2^{0.054n})$. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 107–124. Springer, Heidelberg (2011). doi:10.1007/978-3-642-25385-0_6
32. May, A., Ozerov, I.: On computing nearest neighbors with applications to decoding of binary linear codes. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 203–228. Springer, Heidelberg (2015). doi:10.1007/978-3-662-46800-5_9

33. Meurer, A.: A coding-theoretic approach to cryptanalysis. Dissertation thesis, Universität Bochum Ruhr, Novenber 2012
34. Niebuhr, R., Persichetti, E., Cayrel, P.-L., Bulygin, S., Buchmann, J.: On lower bounds for information set decoding over $\mathbb{F}_q$ and on the effect of partial knowledge
35. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Probl. Control Inf. Theor. **15**, 159–166 (1986)
36. Persichetti, E.: Compact McEliece keys based on quasi-dyadic Srivastava codes. J. Math. Cryptology **6**(2), 149–169 (2012)
37. Peters, C.: Information-set decoding for linear codes over $\mathbb{F}_q$. Cryptology ePrint Archive 2009/589 (2009)
38. Prange, E.: The use of Information-Sets in decoding cyclic codes. IEEE Trans. **IT–8**, S5–S9 (1962)
39. Repka, M., Zajac, P.: Overview of the McEliece cryptosystem and its security. Tatra Mountains Math. Publ. **60**, 57–83 (2014). doi:10.2478/tmmp-2014-0025
40. Stern, J.: A method for finding codewords of small weight. In: Cohen, G., Wolfmann, J. (eds.) Coding Theory 1988. LNCS, vol. 388, pp. 106–113. Springer, Heidelberg (1989). doi:10.1007/BFb0019850
41. Umana, V.G., Leander, G.: Practical key recovery attacks on two McEliece variants. In: International Conference on Symbolic Computation and Cryptography SCC 2010, vol. 2010, p. 62 (2010)