On lower bounds for Information Set Decoding over $\mathbb{F}_q$

Robert Niebuhr[1], Pierre-Louis Cayrel[2], Stanislav Bulygin[2], and Johannes Buchmann[1]

[1] Technische Universität Darmstadt
Fachbereich Informatik
Kryptographie und Computeralgebra,
Hochschulstraße 10
64289 Darmstadt
Germany
{rniebuhr, buchmann}@cdc.informatik.tu-darmstadt.de
[2] CASED – Center for Advanced Security Research Darmstadt,
Mornewegstrasse, 32
64293 Darmstadt
Germany
{pierre-louis.cayrel, stanislav.bulygin}@cased.de

Abstract. Code-based cryptosystems are promising candidates for post-quantum cryptography. The increasing number of cryptographic schemes that are based on codes over non-binary fields $\mathbb{F}_q$ requires an analysis of their security. Information Set Decoding (ISD) is one of the most important generic attacks against code-based cryptosystems. We give lower bounds for ISD over $\mathbb{F}_q$, thereby anticipating future software and hardware improvements. Our results allow to calculate conservative parameters for cryptographic applications.

Key words: Information Set Decoding, lower bounds, codes, post quantum, cryptography.

Introduction

Error-correcting codes have been applied in cryptography for at least three decades since R. McEliece published his paper in 1978. It has received much attention as it is a promising candidate for post-quantum cryptography. McEliece used the class of binary Goppa codes for his construction, and most other schemes published since then have also been using binary codes. However, in recent years, many new proposals use codes over larger fields $\mathbb{F}_q$, mostly in an attempt to reduce the size of the public and private keys. Two examples that received a lot of attention are quasi-cyclic codes [2] by Berger at al., and quasi-dyadic codes [10] (Misoczki-Barreto). The security, however, is not as well understand for q-ary codes than for binary ones: Faugère et al. [6] published an attack which broke these two cryptosystems for several sets of parameters. This makes it important to analyze the complexity of attacks against code-based cryptosystems over larger fields $\mathbb{F}_q$.

The two most important types of attacks against code-based cryptosystems are structural attacks and decoding attacks. Structural attacks exploit structural weaknesses in the construction, and often they attempt to recover the private key. Decoding attacks are used to decode a given cipher text. In this paper, we will not consider structural attacks, since they are restricted to certain constructions or classes of codes.

Information Set Decoding (ISD) is one of the most important generic decoding attacks, and it is the most efficient against many schemes. Over the years, there have been many improvements and generalizations of this attack, e.g. Lee-Brickell [8], Stern [13], Canteaut-Chabaud [5], Bernstein et al. [4], Finiasz-Sendrier [7], and Peters [11].

Our contribution

The publications of Finiasz-Sendrier and Peters on ISD are the most important for our paper: Finiasz and Sendrier provided lower bounds for binary ISD; Peters described how to generalize Stern's and Lee-Brickell's algorithm to $\mathbb{F}_q$. We make the next step and provides lower bounds for ISD over $\mathbb{F}_q$. We prove these bounds, and also show how to use the structure of $\mathbb{F}_q$ to increase the algorithm efficiency. We compare our lower bounds with the most recent ISD algorithm by Peters, and explain how the runtime difference is a result of our conservative assumptions.

Organization of the paper

In Section 1, we start with a review on coding theory and cryptography over $\mathbb{F}_q$. The subsequent Section 2 presents the Information Set Decoding algorithm and states the lower bounds result. In Section 3, we apply these lower bounds to concrete parameters and compare the results with the most recent algorithm. We conclude in Section 4.

1 Review

1.1 Coding theory over F_q

In general, a linear code $\mathcal{C}$ is a k-dimensional subspace of an n-dimensional vector space over a finite field $\mathbb{F}_q$, where k and n are positive integers with $k \leq n$, and q is a prime power. The error-correcting capability of such a code is the maximum number t of errors for which an efficient decoding algorithm is known. In short, linear codes with these parameters are denoted (n, k, t)-codes.
Another common notation is the *co-dimension* r of a code where $r = n - k$.

Definition 1 (Hamming weight). *The (Hamming) weight* $\mathrm{wt}(x)$ *of a vector x is the number of non-zero entries.*

Definition 2 (Minimum distance). *The (Hamming) distance $d(x, y)$ between two codewords $x, y \in \mathcal{C}$ is defined as the (Hamming) weight of $x - y$. The minimum weight d of a code $\mathcal{C}$ is defined as the minimum distance between any two different codewords, or equivalently as the minimum weight over all non-zero codewords:*

$$d := \min_{\substack{x,y \in \mathcal{C} \\ x \neq y}} d(x, y) = \min_{\substack{c \in \mathcal{C} \\ c \neq 0}} \mathrm{wt}(c).$$

A linear code of length n, dimension k and minimum distance d is called an $[n, k, d]$-code.

Definition 3 (Generator and Parity Check Matrix). *Let $\mathcal{C}$ be a linear code over $\mathbb{F}_q$. A generator matrix G of $\mathcal{C}$ is a matrix whose rows form a basis of $\mathcal{C}$:*

$$\mathcal{C} = \{xG : x \in \mathbb{F}_q^k\}.$$

Two generator matrices generate equivalent codes *if one is obtained from the other by a linear transformation. Therefore, we can write any generator matrix G in* systematic form $G = [I_k|R]$, *which allows a more compact representation.*

A parity check matrix H of $\mathcal{C}$ is defined by

$$\mathcal{C} = \{x \in \mathbb{F}_q^n : Hx^T = 0\}$$

and generates the dual space of $\mathcal{C}$. If $\mathcal{C}$ is generated by $G = [I_k|R]$, then a parity check matrix for $\mathcal{C}$ is $H = [-R^T|I_{n-k}]$ (sometimes H can be transformed so that the identity submatrix is on the left hand side).
For a given parity check matrix H and any vector e, we call s the syndrome *of e with $s^T = He^T$.*

The problems which cryptographic applications rely upon can have different numbers of solutions. For example, public key encryption schemes usually have exactly one solution, while digital signatures often have more than one possible solution. The uniqueness of solutions can be expressed by the Gilbert-Varshamov (GV) bound:

Definition 4 (q-ary Gilbert-Varshamov bound). *Let $\mathcal{C}$ be an (n,k,t) code over $\mathbb{F}_q$, and let $r := n-k$. The q-ary GV bound is the smallest integer t_0 such that*

$$\sum_{i=0}^{t_0} \binom{n}{i}(q-1)^i \geq q^r.$$

For large values of n, the last term dominates the sum, so the condition is often approximated by

$$\binom{n}{t_0}(q-1)^{t_0} \geq q^r.$$

If the number of errors that have to be corrected is smaller than the GV bound, then there is at most one solution. Otherwise, there can be several solutions.

1.2 The syndrome decoding problem and the McEliece PKC

Problem 1. Given a matrix H and a vector s, both over $\mathbb{F}_q$, and a non-negative integer t; find a vector $x \in \mathbb{F}_q$ of weight t such that $Hx^T = s^T$.

This problem was proved to be NP-complete in 1978 [3], but only for binary codes. In 1994, A. Barg proved that this result holds for codes over all finite fields [1, in russian]. See Appendix A for a translation of his proposition.

Many code-based cryptographic schemes are based on the hardness of syndrome decoding. Among these are the McEliece cryptosystem and the CFS signature scheme. The latter, however, is unsuitable for q-ary codes, since it requires codes with a high density (ratio of the number of codewords to the cipher space size), and the density rapidly decreases with increasing field size q. We will therefore briefly describe the McEliece cryptosystem and show how it can be attacked by solving the syndrome decoding problem.

Example: The McEliece PKC The McEliece public-key encryption scheme was presented by R. McEliece in 1978 ([9]). The original scheme uses binary Goppa codes, for which it remains unbroken, but the scheme can be used with any class of codes for which an efficient decoding algorithm is known.

Let G be a generator matrix for a linear (n,k,t)-code over $\mathbb{F}_q$, $\mathcal{D}_G$ a corresponding decoding algorithm. Let P be a random permutation matrix and S an invertible matrix over $\mathbb{F}_q$. These form the private key, while $(\widehat{G}, t)$ is made public, where $\widehat{G} = SPG$.

Encryption: Represent the plaintext as a vector m of length k over $\mathbb{F}_q$, choose a q-ary random error vector e of weight at most t, and compute the ciphertext

$$c = m\widehat{G} + e.$$

Decryption: Compute

$$\widehat{c} = cP^{-1} = mSG + eP^{-1}.$$

As P is a permutation matrix, eP^{-1} has the same weight as e. Therefore, $\mathcal{D}_G$ corrects these errors:

$$mSG = \mathcal{D}_G(\widehat{c})$$

Let $J \subseteq \{1,\ldots,n\}$ be a set such that $G_{\cdot J}$ is invertible, then we can compute the plaintext

$$m = mSG \cdot G_{\cdot J}^{-1} \cdot S^{-1}.$$

Attacking the McEliece PKC Attacking the scheme can be achieved by solving the problem of finding close codewords. Information Set Decoding algorithms typically solve a different problem, they find codewords with small Hamming weight. However, this can be used to find close codewords with only a minor inefficiency [5, page 368] (the dimension of the code is increased by 1). The main idea is the following:

Let $\mathcal{C}$ be an $[n, k, d]$ code over $\mathbb{F}_q$, t an integer with $t < d$, and $x \in \mathbb{F}_q$. Finding a codeword $c \in \mathcal{C}$ with $d(c, x) = t$ is equivalent to finding a codeword of weight t in the code $\mathcal{C}' = \mathcal{C} + \{0, x\}$: If $d(c, x) = t$, then $(x - c) \in \mathcal{C}'$ and $\mathrm{wt}(x - c) = t$. Conversely, any codeword $y \in \mathcal{C}'$ of weight $\mathrm{wt}(y) = t$ cannot be in $\mathcal{C} + \{0\}$, so it must be in $\mathcal{C} + \{x\}$, and $y = x - c$.

2 Lower bounds for Information Set Decoding over $\mathbb{F}_q$

The algorithm we describe here recovers a q-ary error vector. It is a generalization of [7] to codes over $\mathbb{F}_q$. We first describe how to modify the algorithm to work over $\mathbb{F}_q$, then we show how to use the field structure to increase efficiency by a factor of $\sqrt{q-1}$.

In each step, we randomly re-arrange the columns of the parity check matrix H and transform it into the form

$$H = \left(\begin{array}{c|c} I_{n-k-l} & H_1 \\ \hline 0 & H_2 \end{array}\right), \tag{1}$$

where I_{n-k-l} is the identity matrix of size $(n - k - l)$. Usually, the columns are chosen adaptively to guarantee the success of this step. Although this approach could bias the following steps, it has not shown any influence in practice. The variables l and p (see next step) are algorithm parameters optimized for each attack.

The error vector we are looking for has p errors in the column set corresponding to H_1 and H_2, and the remaining $(t - p)$ errors in the first $(n - k - l)$ columns. We first check all possible error patterns of p errors in the last $k + l$ columns such that the sum S of those p columns equals the syndrome s in the last l rows. We do this by searching for collisions between the two sets L_1 and L_2, where

$$L_1 = \{H_2 e^T : e \in W_1\} \tag{2}$$

$$L_2 = \{s - H_2 e^T : e \in W_2\}, \tag{3}$$

where $W_1 \subseteq \mathcal{W}_{k+l;\lfloor p/2 \rfloor;q}$ and $W_2 \subseteq \mathcal{W}_{k+l;\lceil p/2 \rceil;q}$ are given to the algorithm, and $\mathcal{W}_{k+l;p;q}$ is the set of all q-ary words of length $k + l$ and weight p. Writing $e = [e' | e_1 + e_2]$ and $s = [s_1 | s_2]$ with s_2 of length l, this means we search for vectors e_1 and e_2 of weight $\lfloor p/2 \rfloor$ and $\lceil p/2 \rceil$, respectively, such that

$$H_2 \cdot [e_1 + e_2]^T = s_2^T.$$

If this succeeds, we compute the difference $S - s$; if this does not have weight $t - p$, the algorithm restarts. Otherwise, the non-zero entries correspond to the remaining $t - p$ errors:

$$\begin{aligned} He^T &= \left(\begin{array}{c|c} I_{n-k-l} & H_1 \\ \hline 0 & H_2 \end{array}\right) \begin{pmatrix} e' \\ e_1 + e_2 \end{pmatrix} \\ &= \begin{pmatrix} I_{n-k-l} \cdot e'^T + H_1 \cdot (e_1 + e_2)^T \\ H_2 \cdot (e_1 + e_2)^T \end{pmatrix} \\ &= \begin{pmatrix} I_{n-k-l} \cdot e'^T \\ 0 \end{pmatrix} + S \\ &\overset{!}{=} \begin{pmatrix} s_1^T \\ s_2^T \end{pmatrix} \end{aligned}$$

Therefore, we have

$$I_{n-k-l} \cdot e'^T = s_1^T - H_1 \cdot (e_1 + e_2)^T,$$

revealing the remaining columns of e.

Using the field structure We can use the field structure of $\mathbb{F}_q$ to increase the algorithm efficiency. Note that for all vectors e such that $He^T = s^T$, there are $q-1$ pairwise different vectors e' such that $He'^T = as^T$ for some $a \in \mathbb{F}_q \backslash \{0\}$, namely $e' = ae$. Clearly, if we find such an e', we can calculate e which solves the syndrome decoding problem. We can modify the algorithm to allow it to find these vectors e' as well, thereby increasing the fraction of error vectors that are (implicitly) tested in each iteration by a factor of $q-1$ (see the Appendix for a detailed description).
Since this fraction is calculated using $|W_1| \cdot |W_2|$, we can also keep the fraction constant and decrease the size of the sets W_i by a factor of $\sqrt{q-1}$ each. As the work factor in each iteration of the algorithm is linear in $|W_1|+|W_2|$, this increases the algorithm efficiency by a factor of $\sqrt{q-1}$.

A simple way to decrease the size of the sets W_i is to redefine them as follows. For any vector a over $\mathbb{F}_q$, let us denote its first non-zero entry by $a(0) \in \mathbb{F}_q \backslash \{0\}$, and let

$$W_1' \subseteq \{e \in \mathcal{W}_{k+l;\lfloor p/2 \rfloor;q} : e(0) = 1\} \tag{4}$$

$$L_1' = \left\{(H_2 e^T)((H_2 e^T)(0))^{-1} : e \in W_1'\right\} \tag{5}$$

$$L_2' = \left\{(s - H_2 e^T)((s - H_2 e^T)(0))^{-1} : e \in W_2\right\}. \tag{6}$$

The algorithm thus works as follows:

Algorithm 1 Information Set Decoding over $\mathbb{F}_q$

Parameters:

- Code parameters: Integers n, $r = n - k$ and t, and a finite field $\mathbb{F}_q$
- Algorithm parameters: Two integers $p > 0$ and $l > 0$, and two sets $W_1 \subseteq \{e \in \mathcal{W}_{k+l;\lfloor p/2 \rfloor;q} : e(0) = 1\}$ and $W_2 \subseteq \mathcal{W}_{k+l;\lceil p/2 \rceil;q}$

Remark: The function $h_l(x)$ returns the last l bits of the vector $x \in \mathbb{F}_q^n$.

Input: Matrix $H_0 \in \mathbb{F}_q^{r \times n}$ and a vector $s_0 \in \mathbb{F}_q^r$

```
Repeat                                                      (MAIN LOOP)
    $P \leftarrow$ random $n \times n$ permutation matrix
    $(H, U) \leftarrow$ PGElim($H_0 P$)             //partial Gauss elimination as in (1)
    $s \leftarrow s_0 U^T$
    for all $e_1 \in W_1$
        $i \leftarrow h_l(He_1^T / y)$                                     (ISD 1)
        write($e_1, i$)                  //store $e$ in some data structure at index $i$
    for all $e_2 \in W_2$
        $i \leftarrow h_l((s^T - He_2^T)/z)$                               (ISD 2)
        $S \leftarrow$ read($i$)                   //extract the elements stored at index $i$
        for all $e_1 \in S$
            if wt($s^T - H(e_1 + e_2)^T) = t - p$                          (ISD 3)
                return $(P, e_1 z/y + e_2)$,                                (SUCCESS)
```

where $y := (He_1^T)(0)$ and $z := (s - He_2^T)(0)$ are notational shortcuts.

Proposition 1. *If $\binom{n}{t}(q-1)^t < q^r$ (single solution), or if $\binom{n}{t}(q-1)^t \geq q^r$ (multiple solutions) and $\binom{r}{t-p}\binom{k}{p}(q-1)^t \ll q^r$, a lower bound for the expected cost (in binary operations) of the algorithm is*

$$WF_{qISD}(n,r,t,p,q) = \min_{p} \frac{1}{\sqrt{q-1}} \cdot \frac{2l \min\left(\binom{n}{t}(q-1)^t, q^r\right)}{\lambda_q \binom{r-l}{t-p}\binom{k+l}{p}(q-1)^t} \cdot \sqrt{\binom{k+l}{p}(q-1)^p}$$

with $l = \log_q\left(K_q \lambda_q \sqrt{\binom{k}{p}(q-1)^{p-1}} \cdot \ln(q)/2\right)$ *and* $\lambda_q = 1 - \exp(-1) \approx 0.63$.
An exception is $p = 0$ *where we cannot gain a factor of* $\sqrt{q-1}$*, hence*

$$WF_{qISD}(n,r,t,0,q) = \frac{\binom{n}{t}}{\binom{r}{t}}$$

If $\binom{n}{t}(q-1)^t \geq q^r$ *and* $\binom{r}{t-p}\binom{k}{p}(q-1)^t \geq q^r$*, the expected cost is*

$$WF_{qISD} \approx \min_{p} \frac{2lq^{r/2}}{\sqrt{\binom{r-l}{t-p}(q-1)^{t-p}}}$$

with $l \approx \log_q\left(K_{t-p}\frac{q^{r/2}}{\sqrt{\binom{r}{t-p}(q-1)^{t-p}}} \cdot \ln(q)/2\right)$.

Remark 1. A realistic value for K_q is $K_q = 2t$. This will be used for the parameters in Section 3.

Remark 2. In the algorithm described above, all computations are done over $\mathbb{F}_q$, so the complexity also depends on the implementation of q-ary arithmetic. A naïve implementation yields an additional factor of $\log_2(q)$ for addition and $\log_2^2(q)$ for multiplication. There are several techniques to improve this, e.g. by lifting to $\mathbb{Z}[x]$ (for large q) or by precomputing exp and log tables (for small q). Especially for small q, this allows to make q-ary arithmetic nearly as fast as binary, so in order to gain conservative estimates, we will neglect this factor.

Remark 3. The total work factor is the product of the number of iterations by the work factor per iterations. In practice, the latter is essentially the sum of a matrix multiplication (with the permutation matrix), the Gaussian elimination, and the search for collisions between L_1' and L_2'. Compared with the binary case, the Gaussian elimination is slower in the q-ary case, because every row has to be divided by the pivot entry. However, since the matrix multiplication and the Gaussian elimination are much faster than the collision search, we do not allocate any cost to them.

3 Results

In [11], the author shows how to extend Lee-Brickell's and Stern's algorithms to codes over $\mathbb{F}_q$. The website [12] lists the work factor of this algorithm against several parameters. We use the same parameters and compare these results with our lower bound.

Table from C. Peters [12], containing parameters for quasi-cyclic [2] and quasi-dyadic [10] codes:

Code parameters				Claimed	$\log_2$(#bit ops)	Lower bound
q	n	k	w	security level	(from [12])	$\log_2$(#bit ops)
256	459	255	50	80	81.93	65.05
256	510	306	50	90	89.43	72.93
256	612	408	50	100	102.41	86.49
256	765	510	50	120	101.58	85.14
1024	450	225	56	80	83.89	62.81
1024	558	279	63	90	91.10	69.81
1024	744	372	54	110	81.01	58.39
4	2560	1536	128	128	181.86	173.23
16	1408	896	128	128	210.61	201.60
256	640	512	64	102	184.20	171.88
256	768	512	128	136	255.43	243.00
256	1024	512	256	168	331.25	318.61
2	2304	1281	64	80	83.38	76.86
2	3584	1536	128	112	112.17	105.34
2	4096	2048	128	128	136.47	129.05
2	7168	3073	256	192	215.91	206.91
2	8192	4096	256	256	265.01	254.16

For the algorithm from [11] as well as for our lower bound algorithm, the expected number of binary operations is the product of the number of iterations by the number of binary operations in each iteration. While the former factor is the same for both algorithms or even a little higher for our algorithm, the lower bound for the number of operations per iteration is much smaller, which results in the difference between these algorithms.

3.1 Difference in the number of operations per iteration

The number of operations per iteration for the first algorithm is the sum of three steps:

1. Reusing parts of information sets and perform precomputations
2. Compute sums of p rows to calculate He^T
3. For each collision (e_1, e_2), check if $\mathrm{wt}(s^T - H(e_1+e_2)^T) = t - p$

To compare the cost of these steps with that used for our lower bound, we calculate all values for the $(450, 225, 56)$ parameter set over $\mathbb{F}_{1024}$. For this set, using $p = 1$, $l = 2$, $m = 1$, $c = 40$ and $r = 1$ (the last three are parameters specific for the first algorithm), we calculate a total cost of the first algorithm of $2^{76.6}$, which consists of 2^{52} iterations of $2^{24.6}$ operations each.

Precomputations The cost of the first step is given in [11] as

$$(n-1)\left((k-1)\left(1-\frac{1}{q^r}\right)+(q^r-r)\right)\frac{c}{r},$$

where c and r are algorithm parameters (i.e. r is *not* the co-dimension of the code). For these parameters, this amounts to $2^{24.4}$ operations, so it is the most expensive step.
Our algorithm does not use precomputation, so we allocate no cost.

Compute sums of p rows to calculate He^T The cost of this step for the first algorithm is

$$\left(\left(\frac{k}{2}-p+1\right)+2\binom{k/2}{p}(q-1)^p\right)l.$$

For the parameters given above, this step adds $2^{18.8}$ operations.

Our algorithm allocates to this step a cost of

$$l|W_1'| + l|W_2| = 2l\sqrt{\binom{k+l}{p}(q-1)^{p-1}}.$$

We make this optimistic assumption[3] for the cost of a matrix-vector multiplication to anticipate further software and hardware improvements for this operation. The result is 2^6 operations in this case.

Check collisions The first algorithm allocates a cost of

$$\frac{q}{q-1}(w-2p)2p\left(1+\frac{q-2}{q-1}\right)\frac{\binom{k/2}{p}^2(q-1)^{2p}}{q^l}$$

to this step. For our set of parameters, this equals $2^{21.4}$ operations.

In our algorithm, we expect the number of collisions to be

$$\frac{\lambda_q|W_1'|\cdot|W_2|}{q^l} = \frac{\lambda_q\binom{k+l}{p}(q-1)^{p-1}}{q^l}.$$

The cost K_q to check each collision is taken to be $K_q = 2t$. Since the expected number of collisions per iteration is very small, the expected cost per iteration is < 1.

Some of the assumptions above may seem fairly optimistic. However, we find that necessary since we want to establish conservative lower bounds.

4 Conclusion and Outlook

In this paper, we have presented and proved lower bounds for Information Set Decoding algorithms over $\mathbb{F}_q$. Part of the result is a modification of the algorithms from [7] which allows to increase the efficiency of the algorithm by a factor of $\sqrt{q-1}$.

It can be seen from the table in Section 3 that over $\mathbb{F}_2$ the efficiency of concrete algorithms is not far from the lower bound, while over larger fields the gap is wider. We propose to further investigate improvements over $\mathbb{F}_q$ to decrease the size of this gap.

Also, in some situations an attacker has partial knowledge of the error vector. For example, in the FSB hash function it is known that the solution e (of $He^T = s^T$) is a regular word, that means that each block of size n/t has weight 1. It should be analyzed how partial knowledge of the solution can increase the efficiency of attacks in order to better estimate the security of cryptographic schemes.

Acknowledgements

References

[1] Barg, S.: Some New NP-Complete Coding Problems. In: *Probl. Peredachi Inf.* 30 (1994), S. 23–28. – (in Russian)

[3] From the cryptanalysts point of view.

[2] Berger, T. P. ; Cayrel, P.-L. ; Gaborit, P. ; Otmani, A.: Reducing Key Length of the McEliece Cryptosystem. In: *AFRICACRYPT* Bd. 5580, Springer, 2009 (Lecture Notes in Computer Science), S. 77–97

[3] Berlekamp, E. ; McEliece, R. ; Tilborg, H. van: On the inherent intractability of certain coding problems. In: *IEEE Trans. Inform. Theory* 24 (1978), Nr. 3, S. 384–386

[4] Bernstein, D. J. ; Lange, T. ; Peters, C.: Attacking and defending the McEliece cryptosystem. In: *PQCrypto '08: Proceedings of the 2nd International Workshop on Post-Quantum Cryptography*. Berlin, Heidelberg : Springer-Verlag, 2008. – ISBN 978–3–540–88402–6, S. 31–46

[5] Canteaut, A. ; Chabaud, F.: 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. In: *IEEE Transactions on Information Theory* 44 (1998), Nr. 1, S. 367–378

[6] Faugère, J.-C. ; Otmani, A. ; Perret, L. ; Tillich, J.-P.: *Algebraic Cryptanalysis of McEliece Variants with Compact Keys*. 2009. – (preprint)

[7] Finiasz, M. ; Sendrier, N.: Security Bounds for the Design of Code-based Cryptosystems. In: *Advances in Cryptology – Asiacrypt'2009*, 2009. – http://eprint.iacr.org/2009/414.pdf

[8] Lee, P.J. ; Brickell, E.F.: An observation on the security of McEliece's public-key cryptosystem. In: *EUROCRYPT '88, Lect. Notes in CS*, 1988, S. 275–280

[9] McEliece, R.J.: A Public-key cryptosystem based on algebraic coding theory. In: *DNS Progress Report* (1978), S. 114–116

[10] Misoczki, R. ; Barreto, P. S. L. M.: Compact McEliece Keys from Goppa Codes. In: *Selected Areas in Cryptography, 16th Annual International Workshop, SAC 2009* Bd. 5867, Springer, 2009 (Lecture Notes in Computer Science)

[11] Peters, C.: *Information-set decoding for linear codes over* $\mathbb{F}_q$. Cryptology ePrint Archive, Report 2009/589, 2009. – http://eprint.iacr.org/

[12] Peters, C.: *Iteration and operation count for information-set decoding over* $\mathbb{F}_q$. Jan 2010. – `http://www.win.tue.nl/∼cpeters/isdfq.html`

[13] Stern, J.: A method for finding codewords of small weight. In: *Proc. of Coding Theory and Applications*, 1989, S. 106–113

A Translation of Barg's proposition and proof

Proposition 2. *The problem P_1 [Coset Weights] is NP-complete over an arbitrary alphabet.*

The proof reduces the 3-dimensional matching problem to the problem of Coset Weights and is analogous to the proof for the binary case in [3].

Proof. Let us construct a $3t \times |U|$-incidence matrix H of the set U. The columns of the matrix correspond to triples $u_i \in U$. Let $u_i = (u_{i_1}, u_{i_2}, u_{i_3})$. Then the i-th column of the matrix H contains $3t-3$ zeroes, and 3 ones at positions u_{i_1}, $t+u_{i_1}$, $2t+u_{i_3}$. In order to construct a concrete problem from P_1 that corresponds to a problem of 3-dimensional matching, set $y = (1, 1, \dots, 1)$ ad $w = t$. Then, if in U there is a subset satisfying the 3-dimensional matching condition and x is its characteristic vector, then $Hx^T = y^T$. Conversely, if this equality is satisfied for some vector x of weight $\leq t$, then it is clear that $\mathrm{wt}(x) = t$. This means that for each i with $0 \leq i \leq 3t$, the sum $y_i = \sum_{j=1}^{|U|} x_j h_{ij}$ contains only one non-zero summand, say i_0, and $x_{i_0} = 1$, i.e. x defines a 3-dimensional matching. □

B Proof of Proposition 1

Except for the additional factor of $1/\sqrt{q-1}$, the proof is similar to that in [7]. We will use the same approach and focus on the differences. As above, let $y(0)$ denote the first non-zero entry of vector $y \in \mathbb{F}_q^n \backslash \{0\}$.

B.1 Efficiency improvement using the field structure of $\mathbb{F}_q$

The step of the algorithm that can be made more efficient using the field structure of $\mathbb{F}_q$ is the search for a pair (e_1, e_2) such that $e_1 \in \mathcal{W}_{k+l;\lfloor p/2 \rfloor;q}$, $e_2 \in \mathcal{W}_{k+l;\lceil p/2 \rceil;q}$ and

$$He_1^T = s^T - He_2^T,$$

where $\mathcal{W}_{k+l;p;q}$ is the set of all q-ary words of length $k+l$ and weight p.

Let W_1', W_2, L_1' and L_2' be defined as in (4)-(6). First note that for any pair (e_1, e_2) and all non-zero values $y \in \mathbb{F}_q$, we have

$$He_1^T = s^T - He_2^T \Leftrightarrow (He_1^T)y^{-1} = (s^T - He_2^T)y^{-1}.$$

Instead of $s^T - He_2^T$, we can store $(s^T - He_2^T)((s^T - He_2^T)(0))^{-1}$ in L_2' and $(He_1^T)(He_1^T(0))^{-1}$ in L_1'. The list L_1', however, would contain every entry exactly $(q-1)$ times, since for every $y \in \mathbb{F}_q \backslash \{0\}$, e_1 and ye_1 yield the same entry. Therefore, we can generate the first list using only vectors e_1 whose first non-zero entry is 1.

To see that there is exactly one collision between L_1' and L_2' for every solution of the problem, let (e_1, e_2) be a pair found by our algorithm. Let $y = He_1^T(0)$ and $z = (s^T - He_2^T)(0)$. Then we have

$$(He_1^T)y^{-1} = (s^T - He_2^T)z^{-1},$$

and therefore $(e_1 z y^{-1}, e_2)$ is a solution to the problem.

Conversely, let (e_1, e_2) be a solution to the problem, i.e. $He_1^T + He_2^T = s^T$. We want to show that there exists a collision between L_1' and L_2' which corresponds to this solution. Let $y = He_1^T(0)$ and $z = (s^T - He_2^T)(0)$. Since $He_1^T = s^T - He_2^T$, we have

$$(He_1^T)y^{-1} = (s^T - He_2^T)z^{-1}. \tag{7}$$

As we did not limit the set W_2, the right hand side of equation (7) belongs to L_2'.

Let $x = e_1(0)$. The first non-zero entry of $e_1' = e_1 x^{-1}$ is 1, so it was used to calculate one member of L_1'. As $He_1'^T(0) = (H(e_1x^{-1})^T)(0) = yx^{-1}$,

$$(He_1'^T)((He_1'^T)(0))^{-1} = (H(e_1x^{-1})^T)(yx^{-1})^{-1} = (He_1^T)y^{-1}.$$

Therefore, the left hand side of equation (7) belongs to L_1'.
Since $z = y$, this collision between L_1' and L_2' corresponds to the solution (e_1, e_2).

Obviously, this improvement can only applied if $p > 0$, i.e. if there actually is a search for collisions. If $p = 0$, we are simply trying to find a permutation which shifts all error positions into the first r positions of s, so the runtime is the inverse of the probability P_0 of this event with $P_0 = \binom{r}{t}/\binom{n}{t}$. For the rest of this section we assume $p > 0$.

B.2 Cost of the algorithm

In most cases, the value of t will be smaller than the GV bound, and we expect the algorithm to require many iterations. In that case, in one iteration of our Main Loop, we expect to test a fraction $\lambda_q(z) = 1 - \exp(z_q)$ of vectors in $\mathcal{W}_{k+l;p;q}$, where

$$z_q = \frac{|W_1'| \cdot |W_2|}{\binom{k+l}{p}(q-1)^{p-1}}. \tag{8}$$

The success probability of each pair (e_1, e_2) is the number of pairs matching the syndrome in the last l rows, divided by the total number of possible values of He with $e \in \mathcal{W}_{k+l;p;q}$. Depending on the code parameters, the latter is either given by the number of error patterns or by the number of syndromes:

$$P_q = \frac{\lambda_q(z_q)\binom{r-l}{t-p}(q-1)^{t-p}}{\min\left(\binom{n}{t}(q-1)^t, q^r\right)}$$

The success probability in one iteration of Main Loop is hence:

$$\begin{aligned} P_{p;q}(l) &= 1 - (1 - P_q)^{\binom{k+l}{p}(q-1)^p} \\ &\approx 1 - \exp(-P_q \cdot \binom{k+l}{p}(q-1)^p) \\ &= 1 - \exp\left(-\frac{\lambda_q(z_q)}{N_{p;q}(l)}\right), \end{aligned}$$

where

$$N_{p;q}(l) = \frac{\min\left(\binom{n}{t}(q-1)^t, q^r\right)}{\binom{r-l}{t-p}\binom{k+l}{p}(q-1)^t}.$$

For small $P_{p;q}(l)$, the cost of the algorithm can be calculated approximately as

$$\frac{N_{p;q}(l)}{\lambda_q(z_q)} \cdot \left(l|W_1'| + l|W_2| + K_q \frac{\lambda_q(z_q)\binom{k+l}{p}(q-1)^{p-1}}{q^l}\right),$$

which is the approximate number of iterations times the number of operations per iteration. K_q is the expected cost to perform the check $\mathrm{wt}(s^T - H(e_1 + e_2)^T) = t - p$.

It is easy to see that we minimize this formula by choosing $|W_1'| = |W_2|$,

$$N_{p;q}(l) \cdot \left(2l\frac{|W_1'|}{\lambda_q(z_q)} + K_q \frac{\binom{k+l}{p}(q-1)^{p-1}}{q^l}\right)$$

Using (8), we get

$$N_{p;q}(l) \cdot \left(2l\frac{\sqrt{z_q}}{\lambda_q(z_q)}\sqrt{\binom{k+l}{p}(q-1)^{p-1}} + K_q \frac{\binom{k+l}{p}(q-1)^{p-1}}{q^l}\right)$$

Analytically, the optimal value for z_q is $z \approx 1.25$, but $z_q = 1$ is very close to optimal. Hence we choose $z_q = 1$, set $\lambda_q = \lambda_q(1) = 1 - e^{-1}$ and use (8),

$$N_{p;q}(l)\sqrt{\binom{k+l}{p}(q-1)^{p-1}}\frac{2}{\lambda_q} \cdot \left(l + \frac{K_q\lambda_q}{2} \cdot \frac{\sqrt{\binom{k+l}{p}(q-1)^{p-1}}}{q^l}\right).$$

The optimal value for l can be approximated by $l = \log_q\left(K_q\lambda_q\sqrt{\binom{k+l}{p}(q-1)^{p-1}} \cdot \ln(q)/2\right)$. In practice, we use $l \approx \log_q\left(K_q\lambda_q\sqrt{\binom{k}{p}(q-1)^{p-1}} \cdot \ln(q)/2\right)$. For small values of q, the factor $(\ln(q)/2)$ can be neglected. Hence the cost is

$$\frac{1}{\sqrt{q-1}} \cdot \frac{2l\min\left(\binom{n}{t}(q-1)^t, q^r\right)}{\lambda_q\binom{r-l}{t-p}\binom{k+l}{p}(q-1)^t} \cdot \sqrt{\binom{k+l}{p}(q-1)^p}$$

Minimizing over p gives the result.

Now consider the case where $\binom{r}{t-p}\binom{k}{p}(q-1)^t \geq q^r$. Then the Main Loop is likely to succeed after a single iteration. This corresponds to the birthday algorithm described in [7], which states

$$\mathrm{WF}_{\mathrm{BA}} \approx \frac{2}{\sqrt{P}} \cdot \left(l + \frac{K_0}{2\sqrt{P}2^l} \right).$$

In the q-ary case this formula becomes

$$\mathrm{WF}_{\mathrm{qBA}} \approx \frac{2}{\sqrt{P}} \cdot \left(l + \frac{K_0}{2\sqrt{P}q^l} \right).$$

Easy analysis shows that the optimal value for l is

$$l = \log_q \left(\frac{\ln(q) K_0}{2\sqrt{P}} \right).$$

Applying this in our case with K_{t-p} instead of K_0 (since K_0 is the cost of the third step in the algorithm of [7], which is K_{t-p} when applied in the case of ISD), using

$$P = P_q \approx \frac{\binom{r-l}{t-p}(q-1)^{t-p}}{q^r},$$

and minimizing over p yields the lower bound result:

$$\mathrm{WF}_{\mathrm{qISD}} \approx \frac{2lq^{r/2}}{\sqrt{\binom{r-l}{t-p}(q-1)^{t-p}}}$$

with $l \approx \log_q \left(K_{t-p} \frac{q^{r/2}}{\sqrt{\binom{r-l}{t-p}(q-1)^{t-p}}} \cdot \ln(q)/2 \right)$.