A Non Asymptotic Analysis of Information Set Decoding

Yann Hamdaoui and Nicolas Sendrier

INRIA Paris-Rocquencourt, Project-Team SECRET
{yann.hamdaoui,nicolas.sendrier}@inria.fr

Abstract. We propose here a non asymptotic complexity analysis of some variants of information set decoding. In particular, we give this analysis for the two recent variants – published by May, Meurer and Thomae in 2011 and by Becker, Joux, May and Meurer in 2012 – for which only an asymptotic analysis was available. The purpose is to provide a simple and accurate estimate of the complexity to facilitate the paramater selection for code-based cryptosystems. We implemented those estimates and give a comparison at the end of the paper.

Notation:

- $\mathcal{S}_n(\mathbf{0}, w)$ is the radius w sphere centered in $\mathbf{0}$ in the Hamming space $\{0,1\}^n$.
- $|X|$ denotes the cardinality of the set X.

1 Introduction: the Decoding Problem in Cryptology

The security of code-based cryptography heavily relies on the hardness of decoding in a random linear code. The computational syndrome decoding problem is NP-hard and is conjectured difficult in the average case.

Problem 1 (Computational Syndrome Decoding - CSD). *Given a matrix $H \in \{0,1\}^{r\times n}$, a word $s \in \{0,1\}^r$, and an integer $w > 0$, find $e \in \{0,1\}^n$ of Hamming weight $\leq w$ such that $He = s$.*

We will denote $\mathrm{CSD}(H, s, w)$ the above problem and the set of its solutions. Decoding is one of the prominent algorithmic problems in coding theory for more than fifty years. So far, no subexponential algorithm is known which correct a constant proportion of errors in a linear code. Code-based cryptography has been developed on that ground and for many code-based cryptosystems, public-key encryption [18, 20] and digital signature [10], zero-knowledge protocols based on codes [28, 29, 14], hash-function [1], PRNG and stream ciphers [13, 15] and many others, decoding is the most threatening attack and therefore is a key point in the parameter selection.

Generic Decoding Algorithms. The most ancient technique for addressing CSD in cryptology is Information Set Decoding (ISD). It can be traced back to Prange [24]. The variants useful today in cryptology all derive more or less from Stern's [27] or Dumers's [11] algorithms. Following [6, 23] those variants are sometimes refered to as collision decoding. It was implemented (with various improvements) in [8] then in [5] which reports the first successful attack on the original parameter set. General lower bounds were proposed [12]. Several recent works have provided asymptotic improvements [6, 17, 4].

The other main technique is the Generalized Birthday Algorithm (GBA) [30] (order 2 GBA was previously published in [7]). The first use of GBA for decoding was proposed in [9] for attacking an early version of FSB [2]. It is sometimes faster than ISD. Let us also mention that when addressing *multiple instances* there are possible improvement, either with GBA (Bleichenbacher, unpublished, reported in [22]) or with ISD [26].

The security of the various code-based cryptographic primitives corresponds to a wide range of parameters for the CSD problem. To determine which attack is the most efficient, one should compare the error weight w with the Gilbert-Varshamov distance d_0 (which is a function of the code length and size). *For a single instance*, the situation is the following: (1) when $w < d_0$ (for encryption schemes) ISD is always better, (2) when $w \approx d_0$ (for ZK-protocols, digital signature, stream cipher), the best attack is also ISD, and (3) when $w > d_0$ (for hashing) the best attack is either ISD or GBA (with no easy rule to predict which is the best). Let us also mention that $w > r/4$ is insecure because Saarinen's attack [25]. In the current work, we are concerned with the case $w < d_0$, that is mainly McEliece and Niederreiter encryption schemes.

2 A General Framework for Information Set Decoding Algorithms

All known variants of collision decoding [27, 11, 8, 6, 17, 4] can be described within a simple framework similar to the one presented in [12]. The problem to solve is CSD(H_0, s_0, w). The algorithm uses two main parameters p and ℓ (integers) and will repeatedly permute randomly the columns of H_0, perform a (partial) Gaussian elimination[1]

$$UH_0P = \begin{array}{c} \\ \\ \ell \end{array}\begin{array}{|c|c|} \multicolumn{1}{c}{r-\ell} & \multicolumn{1}{c}{k+\ell} \\ \hline \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & H' \\ \hline 0 & H \\ \hline \end{array}\,, \quad Us_0 = \begin{array}{|c|} \hline s' \\ \hline s \\ \hline \end{array} \tag{1}$$

[1] if the first $r - \ell$ columns of H_0P are dependent, which is unlikely, we change P

where U is a non-singular $r \times r$ matrix. We have $e \in \mathrm{CSD}(UH_0P, Us_0, w)$ if and only if $Pe \in \mathrm{CSD}(H_0, s_0, w)$. All algorithms in the class we consider can be described as follows:

```
input: $H_0 \in \{0,1\}^{r\times n}$, $s_0 \in \{0,1\}^r$
repeat until SUCCESS
  1. pick a permutation $P$ randomly and compute $(H, H', s, s')$ as in (1)
  2. compute a set of partial solutions $\mathcal{E} \subset \mathrm{CSD}(H, s, p)$
  3. for all $e \in \mathcal{E}$ if $\mathrm{wt}(e) + \mathrm{wt}(s' + H'e) \leq w$ then SUCCESS
```

The variants will only differ in step **2.**

2.1 Three variants of ISD

Building Blocks. We give in Table 1 the generic building blocks used in all variants. The function `ISD_generic` has to be instantiated with some other func-

Table 1. Information set decoding building blocks

```
Parameters:
  – integers: $n, r, k = n - r, w, p, \ell$
Parameters $p$ and $\ell$ can be chosen to optimize the algorithm.

function ISD_generic
input: $H_0 \in \{0,1\}^{r\times n}$, $s_0 \in \{0,1\}^r$, sub_ISD()
    repeat
        $P \leftarrow$ random_permutation_matrix()    // $n \times n$
        $(H, H', s, s') \leftarrow$ Gauss_elim($H_0P, s_0$)    // as in (1)
        $\mathcal{E} \leftarrow$ sub_ISD($H, s$)    // a subset of $\mathrm{CSD}(H, s, p)$
        for all $e \in \mathcal{E}$
            if $\mathrm{wt}(H'e - s') + \mathrm{wt}(e) \leq w$ then return $(e, P)$    // SUCCESS

function birthday_decoding
input: $H \in \{0,1\}^{b\times c}$, $s \in \{0,1\}^b$, $\mathcal{E}_1 \subset \{0,1\}^c$, $\mathcal{E}_2 \subset \{0,1\}^c$
    $\mathcal{E} \leftarrow \emptyset$
    for all $e_1 \in \mathcal{E}_1$ do $T[He_1] \leftarrow e_1$
    for all $e_2 \in \mathcal{E}_2$
        for all $e_1 \in T[s - He_2]$ do $\mathcal{E} \leftarrow \mathcal{E} \cup \{e_1 + e_2\}$
    return $\mathcal{E}$
```

tion `sub_ISD` which depends on the variant. The function `birthday_decoding` is the elementary building block; given a parity check matrix H, a target syndrome s and two sets of error patterns $\mathcal{E}_1$ and $\mathcal{E}_2$, it computes all the error patterns $e = e_1 + e_2$ such that $He = s$ and $(e_1, e_2) \in \mathcal{E}_1 \times \mathcal{E}_2$.

The Stern-Dumer variant (SD-ISD). Stern's algorithm was the first to use the birthday paradox [27]. It was later improved by Dumer [11] with a slight asymptotic improvement. It is described in Table 2. Ball-collision decoding [6] is similar but more involved with the same asymptotic exponent.

Table 2. SD-ISD algorithm

Parameters:
- integers: $n, r, k = n - r, w, p, \ell$
- $\mathcal{E}_1 \subset \mathcal{S}_{k+\ell}(\mathbf{0}, p/2)$ and $\mathcal{E}_2 \subset \mathcal{S}_{k+\ell}(\mathbf{0}, p/2)$ of cardinality $\binom{(k+\ell)/2}{p/2}$ with disjoint supports. For instance $\mathcal{E}_1$ (respectively $\mathcal{E}_2$) consists of all words of weight $p/2$ with the non-zero positions in the $(k+\ell)/2$ leftmost (respectively rightmost) positions.

Parameters p and ℓ can be chosen to optimize the algorithm.

```
function ISD_SD
input: $H_0 \in \{0,1\}^{r \times n}$, $s_0 \in \{0,1\}^r$
    return ISD_generic($H_0, s_0$, birthday_decoding_SD)
```

```
function birthday_decoding_SD
input: $H \in \{0,1\}^{\ell \times (k+\ell)}$, $s \in \{0,1\}^\ell$
    return birthday_decoding($H, s, \mathcal{E}_1, \mathcal{E}_2$)
```

The May-Meurer-Thomae variant (MMT-ISD). The generalized birthday algorithm [30] can be applied to decoding [9]. Using this in conjonction to the representation technique [16] leads to an asymptotic improvement of ISD [17]. We present in Table 3 a modified version of the original algorithm. In the original algorithm the set A is a singleton. By allowing several $a \in A$ we allow larger values of ℓ_2 and give more flexibility in the search for optimal parameters. A larger ℓ_2 also allows smaller memory requirements with the same algorithmic complexity.

The Becker-Joux-May-Meurer variant (BJMM-ISD). In this variant the birthday decoding of Stern's algorithm (step 2) is replaced by an order 3 generalized birthday decoding. We have eight error sets initially which are merged pairwise in a three-level tree-like algorithm to produce the set $\mathcal{E}$ of candidates of `ISD_generic`. This is done in conjonction with the representation technique and, in addition, at every merging step except the first, the error sums might be filtered to keep only those of smaller weight. The eight sets $\mathcal{E}_1, \ldots, \mathcal{E}_8$ at the last level only contain words of weight $p_2/2$, the four sets $\mathcal{E}_1^{(2)}, \ldots, \mathcal{E}_4^{(2)}$ at the second level only contain words of weight p_2, the two sets $\mathcal{E}_1^{(1)}, \mathcal{E}_2^{(1)}$ at the first level only contain words of weight $\leq p_1 = 2(p_2 - e_2)$ and the final set (the output) only contains words of weight $\leq p = 2(p_1 - e_1)$. We may choose e_1 or e_2 strictly positive. The algorithm is described in Table 4. The function `filter` simply keep words of small weight in a set of words.

Table 3. MMT-ISD algorithm

Parameters:
- integers: $n, r, k = n - r, w, p, \ell, \ell_2$
- $A \subset \{0,1\}^{\ell_2}$
- $\mathcal{E}_i \subset \mathcal{S}_{k+\ell}(\mathbf{0}, p/4), 1 \le i \le 4$ of cardinality $\binom{(k+\ell)/2}{p/4}$. The sets $\mathcal{E}_1$ and $\mathcal{E}_2$ have disjoint supports. The sets of $\mathcal{E}_3$ and $\mathcal{E}_4$ have disjoint supports.

Parameters p, ℓ, ℓ_2, and $|A|$ can be chosen to optimize the algorithm.

```
function ISD_MMT
input: $H_0 \in \{0,1\}^{r\times n}$, $s_0 \in \{0,1\}^r$
    return ISD_generic($H_0, s_0$, birthday_decoding_MMT)
```

```
function birthday_decoding_MMT
input: $H \in \{0,1\}^{\ell\times(k+\ell)}$, $s \in \{0,1\}^\ell$ // H and s decomposed as below in (2)
    $\mathcal{E} \leftarrow \emptyset$
    for all $a \in A$
        $\mathcal{E}_{1,2} \leftarrow$ birthday_decoding($H^{(2)}, a, \mathcal{E}_1, \mathcal{E}_2$)
        $\mathcal{E}_{3,4} \leftarrow$ birthday_decoding($H^{(2)}, s^{(2)} - a, \mathcal{E}_3, \mathcal{E}_4$)
        $\mathcal{E} \leftarrow \mathcal{E} \cup$ birthday_decoding($H^{(1)}, s^{(1)}, \mathcal{E}_{1,2}, \mathcal{E}_{3,4}$)
    return $\mathcal{E}$
```

$$\begin{array}{r|c|}\cline{2-2} H = & H^{(1)} \\ \cline{2-2} \ell_2 & H^{(2)} \\ \cline{2-2}\end{array}\ , \quad s = \begin{array}{|c|}\hline s^{(1)} \\ \hline s^{(2)} \\ \hline\end{array} \tag{2}$$

2.2 Some Comments on the Complexity

We will count complexities in term of column operations. Those operations could be additions, mostly in calls to `birthday_decoding`, or Hamming weight computations, in the final test of `ISD_generic`.

Complexity of `ISD_generic`.

- We denote K_{Gauss} the cost of the partial Gaussian elimination. A naive implementation leads to $K_{Gauss} = (r - \ell)n$ column operations. Fast binary linear algebra [3] lead to $K_{Gauss} = (r - \ell)n/\log_2(r - \ell)$.
  It is possible to reduce this further to $K_{Gauss} = O(n)$ by transposing only one or a few positions at each iterations [21, 8, 5]. The success probability of an iteration decreases. There is a possible gain, but only for instances of small size. We do not consider this possibility here.
- To estimate the success probability of one iteration, we will admit, as it is common in existing literature, that each individual e tested in `ISD_generic`

Table 4. BJMM-ISD algorithm

```
Parameters:
  – integers: $n, r, k = n - r, w, p, \ell, p_1, p_2, r_1, r_2$
  – $\mathcal{E}_i \subset \{0,1\}^{k+\ell}, 1 \le i \le 8$ consisting of words of Hamming weight $p_2/2$.
    All $\mathcal{E}_i$ have cardinality $\binom{(k+\ell)/2}{p_2/2}$. The words of $\mathcal{E}_{2i-1}$ and $\mathcal{E}_{2i}$ have disjoint
    supports.
Parameters $p$, $\ell$, $p_1$, $p_2$, $r_1$, and $r_2$ can be chosen to optimize the algorithm.

function ISD_BJMM
input: $H_0 \in \{0,1\}^{r\times n}$, $s_0 \in \{0,1\}^r$
    return ISD_generic($H_0, s_0,$ birthday_decoding_BJMM)

function birthday_decoding_BJMM
input: $H \in \{0,1\}^{\ell\times(k+\ell)}$, $s \in \{0,1\}^\ell$  // $H$ and $s$ decomposed as below in (3)
    // choose some $s_1 + s_2 + s_3 + s_4 = s^{(2)} \in \{0,1\}^{r_2}$ and $a \in \{0,1\}^{r_1}$
    for all $i \in \{1,2,3,4\}$
        $\mathcal{E}_i^{(2)} \leftarrow$ birthday_decoding($H^{(2)}, s_i, \mathcal{E}_{2i-1}, \mathcal{E}_{2i}$)
    $\bar{\mathcal{E}}_1^{(1)} \leftarrow$ birthday_decoding($H^{(1)}, a, \mathcal{E}_1^{(2)}, \mathcal{E}_2^{(2)}$)
    $\mathcal{E}_1^{(1)} \leftarrow$ filter($\bar{\mathcal{E}}_1^{(1)}, p_1$)                    // keep only words of weight $\le p_1$
    $\bar{\mathcal{E}}_2^{(1)} \leftarrow$ birthday_decoding($H^{(1)}, s^{(1)} - a, \mathcal{E}_3^{(2)}, \mathcal{E}_4^{(2)}$)
    $\mathcal{E}_2^{(1)} \leftarrow$ filter($\bar{\mathcal{E}}_2^{(1)}, p_1$)                    // keep only words of weight $\le p_1$
    $\bar{\mathcal{E}} \leftarrow$ birthday_decoding($H^{(0)}, s^{(0)}, \mathcal{E}_1^{(1)}, \mathcal{E}_2^{(1)}$)
    $\mathcal{E} \leftarrow$ filter($\bar{\mathcal{E}}, p$)                               // keep only words of weight $\le p$
    return $\mathcal{E}$
```

$$H = \begin{array}{r|c|} \hline & \\ & H^{(0)} \\ & \\ \hline r_1 & H^{(1)} \\ \hline r_2 & H^{(2)} \\ \hline \end{array}, \quad s = \begin{array}{|c|} \hline \\ s^{(0)} \\ \\ \hline s^{(1)} \\ \hline s^{(2)} \\ \hline \end{array} \tag{3}$$

leads independently to SUCCESS with probability (see [26])

$$\varepsilon(p,\ell)2^\ell \approx \frac{\binom{r-\ell}{w-p}2^\ell}{\min\left(2^r, \binom{n}{w}\right)}.$$

It follows that the probability of success of one iteration is equal to

$$\mathcal{P}(p,\ell) = 1 - \left(1 - \varepsilon(p,\ell)2^\ell\right)^{|\mathcal{E}|}. \tag{4}$$

The expected value of $|\mathcal{E}|$ will depend on the variant.

– The final test cost at least $|\mathcal{E}|$ column operations. In practice it is more accurate to count the number of matching pairs of the last call to `birthday_decoding`. Filtering out duplicate at this point would have a significant cost and would only save a small amount of tests.

Complexity of `birthday_decoding`. We consider an instance with input $H \in \{0,1\}^{b\times c}$, $s \in \{0,1\}^b$, $\mathcal{E}_1 \subset \{0,1\}^c$, $\mathcal{E}_2 \subset \{0,1\}^c$. There are two fundamental quantities related to this algorithm.

– The number of matching pairs $(e_1, e_2) \in \mathcal{E}_1 \times \mathcal{E}_2$ such that $He_1 + He_2 = s$. This number is expected to be equal to $2^{-b}|\mathcal{E}_1||\mathcal{E}_2|$.
– The size of the output. In fact we will count for some integer w the number of elements of $\mathcal{E}$ of weight w.

$$|\mathcal{E} \cap \mathcal{S}_n(\mathbf{0}, w)| = \mathrm{ImSize}\left(\mu \frac{|\mathcal{E}_1||\mathcal{E}_2|}{2^b}, \frac{\binom{c}{w}}{2^b}\right)$$

where μ is the proportion of words of $\mathcal{E}$ that have weight w and $\mathrm{ImSize}(x, y)$ is the expected size of the image of a random mapping from a set of size x into a set of size y

$$\mathrm{ImSize}(x, y) = y\left(1 - \left(1 - \frac{1}{y}\right)^x\right). \tag{5}$$

It is equal to $\min(x, y)$ up to a small constant (between 0.632 and 1).

In all variants, the calls to `birthday_decoding` have a tree structure and the output sets at one level are used as inputs at the next level. Columns additions will be performed on full columns (*i.e.* colums of H not $H^{(i)}$ in the algorithm description). A careful implementation will store all the syndromes and keep track of partial sums (as in [5]). Finally a good estimate of the complexity will be the sum of the sizes of all initial sets (there are two, four, or eight such sets, depending on the variant) plus the number of matching pairs in all calls of `birthday_decoding` (recall that the matching pairs of the last call have to be tested, so thay have to be counted twice).

3 Non Asymptotic Complexity Estimate

3.1 Some Simplifications

– Using expectations. We will freely replace random variables by their expectations. This is valid for products, not for exponentials.
– In MMT-ISD, we will assume that
  • all the elements in the final set $\mathcal{E}$ have weight p (could be smaller).
– In BJMM-ISD, we will assume that
  • the sets $\mathcal{E}_i^{(2)}$, $1 \leq i \leq 4$ behave as random subsets of $\mathcal{S}_{k+\ell}(\mathbf{0}, p_2)$ (in fact they are balanced),
  • all the elements of $\mathcal{E}_1^{(1)}$ and $\mathcal{E}_2^{(1)}$ have weight p_1 (could be smaller),
  • all the elements of $\mathcal{E}$ have weight p (could be smaller),
  • the cost of filtering is negligible.

Essentially, all those simplifications cannot cost more than a small constant factor, at least when the algorithm parameters are close to the optimal value.

3.2 Stern-Dumer Variant

There is a single call to `birthday_decoding`, the input sets have size $L_0 = |\mathcal{E}_1| = |\mathcal{E}_2| = \binom{(k+\ell)/2}{p/2}$ and there are no duplicates in the output.

$$\mathrm{WF}_{\mathrm{SD}}(n,r,w) = \min_{p,\ell} \frac{1}{\mathcal{P}(p,\ell)} \left(K_{Gauss} + 2L_0 + \frac{2L_0^2}{2^\ell} \right).$$

To compute the minimal value, we have to explore all values of p and ℓ. In practice, when p is fixed, the optimal value of ℓ will be close to the one implicitly defined by $L_0 = \binom{(k+\ell)/2}{p/2} = 2^\ell$ (for which the two rightmost terms of the complexity are equal, see appendix). In addition we limit the search to even values of p. Finally, since the expression is convex in p and ℓ in the region we explore, the search is limited in practice to a very small number of values. Moreover in the region we explore the workfactor is convex in p and in ℓ, this simplifies the search for the optimal value.

3.3 May-Meurer-Thomae Variant

There are initially four sets of error patterns, each of size $L_0 = \binom{(k+\ell)/2}{p/4}$. The first two calls to `birthday_decoding` are made with sets of disjoint supports and lead to an output of expected size $L_0^2 2^{-\ell_2}$ (no duplicates) and the number of matching pairs in the final call is $L_0^4 2^{-\ell-\ell_2}$. This is repeated for all $a \in A$ and finally

$$\mathrm{WF}_{\mathrm{MMT}}(n,r,w) = \min_{p,\ell,\ell_2} \frac{1}{\mathcal{P}(p,\ell)} \left(K_{Gauss} + |A| \left(4L_0 + \frac{2L_0^2}{2^{\ell_2}} + \frac{2L_0^4}{2^{\ell+\ell_2}} \right) \right).$$

The total number of matching pairs in the calls to `birthday_decoding` producing $\mathcal{E}$ is $|A| L_0^4 2^{-\ell-\ell_2}$. In practice, we will choose ℓ_2 and A such that:

- the cost for constructing the initial sets is negligible, that is $2^{\ell_2} \ll L_0$,
- there are no duplicate sums in the matching pairs forming $\mathcal{E}$, that is

$$\frac{|A| L_0^4}{2^{\ell+\ell_2}} \ll \frac{\binom{k+\ell}{p}}{2^\ell} \text{ and thus } |\mathcal{E}| = \frac{|A| L_0^4}{2^{\ell+\ell_2}}.$$

Optimal parameters can be easily estimated, we first neglect the cost of the Gaussian elimination (true if $|A|$ not too small) and assume $\mathcal{P}(p,\ell) \approx \varepsilon(p,\ell) 2^\ell |\mathcal{E}| \approx \varepsilon(p,\ell)|A| L_0^4 2^{-\ell_2}$ (true for most problems of cryptographic interest). The expression of the workfactor simplifies to

$$\mathrm{WF}_{\mathrm{MMT}}(n,r,w) \approx \min_{p,\ell,\ell_2} \frac{2}{\varepsilon(p,\ell)} \left(\frac{1}{L_0^2} + \frac{1}{2^\ell} \right).$$

Interestingly, if A and ℓ_2 are properly chosen, their exact value do not really matter. As before, if we examine the above formula, it will be minimal when

$2^\ell \approx L_0^2 = \binom{(k+\ell)/2}{p/4}^2$. We have to pick $|A|$ and ℓ_2 as large as possible but such that $|A| \ll 2^{\ell_2} \ll 2^{\ell/2}$: a large value of ℓ_2 will decrease the memory requirements and a large value of $|A|$ will better amortize the cost of the Gaussian elimination. In this variant of ISD, the optimal value of p will usually be larger. Consequently the cost of the Gaussian elimination will never dominate and the above analysis is correct in practice. As for the Stern-Dumer variant, the expression of the workfactor is convex in p and ℓ an exploring only a few values will provide the minimum.

3.4 Becker-Joux-May-Meurer Variant

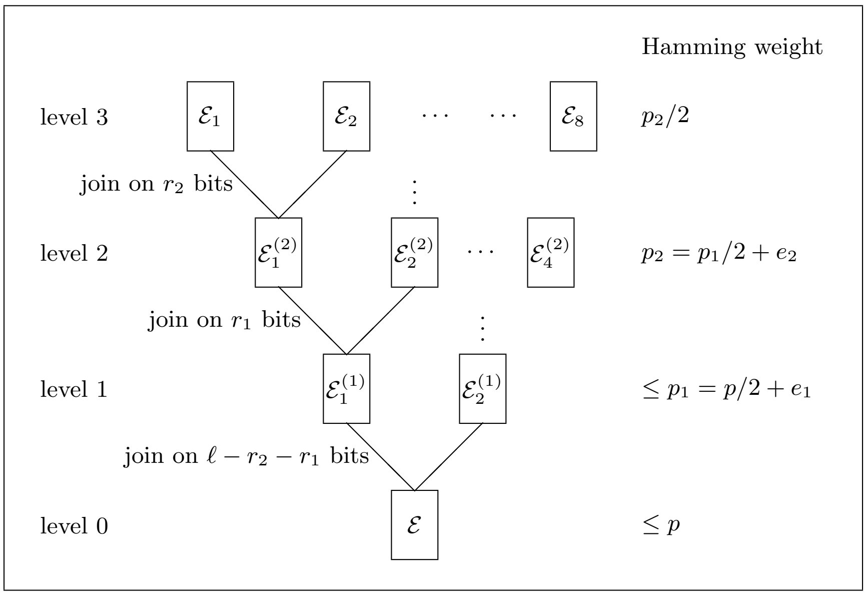

Fig. 1. Tree Structure of BJMM-ISD

We will denote by S_i the size of the sets at level i and by C_i the number of matching pairs when two lists at level i are joined. Notations are given in Table 4 and Figure 1.

$$S_3 = |\mathcal{E}_i|, S_2 = |\mathcal{E}_i^{(2)}|, S_1 = |\mathcal{E}_1^{(1)}| = |\mathcal{E}_2^{(1)}|, S_0 = |\mathcal{E}|$$

$$C_3 = S_2, C_2 = |\bar{\mathcal{E}}_1^{(1)}| = |\bar{\mathcal{E}}_2^{(1)}|, C_1 = |\bar{\mathcal{E}}|$$

$$p_2 = \frac{p_1}{2} + e_2, p_1 = \frac{p}{2} + e_1$$

Level 3. The sets contain words of weight $\frac{p_2}{2}$ on a half-sized support of length $\frac{k+\ell}{2}$. We join such sets pairwise keeping the pairs whose syndromes match on r_2 bits, we have

$$S_3 = \binom{\frac{k+\ell}{2}}{\frac{p_2}{2}} \text{ and } C_3 = \frac{S_3^2}{2^{r_2}}.$$

Level 2. The supports at level 3 are disjoint, so we have no duplicates when building the sets at size 2.

$$S_2 = C_3 \text{ and } C_2 = \frac{S_2^2}{2^{r_1}}.$$

Level 1. From the C_2 matching pairs of level 2, we keep only those with a sum of Hamming weight smaller or equal to $p_1 = 2p_2 - 2e_2$. We denote μ_2 the probability for the sum of two words of weight p_2 to have weight p_1. The min below takes into account possible duplicates (see §2.2).

$$S_1 = \min\left(\mu_2 C_2, \frac{\binom{k+\ell}{p_1}}{2^{r_1+r_2}}\right) \text{ and } C_1 = \frac{S_1^2}{2^{\ell-r_1-r_2}}.$$

Level 0. Similarly, the size of the final set is

$$S_0 = \min\left(\mu_1 C_1, \frac{\binom{k+\ell}{p}}{2^{\ell}}\right)$$

where μ_1 denotes the probability for two words of weight p_1 and length $k+\ell$ to have a sum of weight p.

The values of μ_1 and μ_2 derive from the following result.

Proposition 1. *Two binary words drawn uniformly and independently in $\mathcal{S}_n(\mathbf{0}, w)$ have a sum of weight $2w - 2e$ with probability*

$$\Pr(\text{“}w + w = 2w - 2e\text{”}) = \frac{\binom{w}{e}\binom{n-w}{w-e}}{\binom{n}{w}}.$$

The proof is left to the reader. We deduce that

$$\mu_2 = \frac{\binom{p_2}{e_2}\binom{k+\ell-p_2}{p_2-e_2}}{\binom{k+\ell}{p_2}} \text{ and } \mu_1 = \frac{\binom{p_1}{e_1}\binom{k+\ell-p_1}{p_1-e_1}}{\binom{k+\ell}{p_1}}.$$

Workfactor. The algorithm workfactor is

$$\frac{1}{\mathcal{P}(p,\ell)}\left(K_{Gauss} + 8S_3 + 4C_3 + 2C_2 + 2C_1\right)$$

where

$$\mathcal{P}(p,\ell) = 1 - \left(1 - \varepsilon(p,\ell)2^{\ell}\right)^{S_0} \approx \varepsilon(p,\ell)2^{\ell} S_0.$$

The workfactor has to be minimized according to six parameters $(p, \ell, r_1, r_2, e_1, e_2)$. This could lead to a rather high computational burden, but fortunately it is possible to guess what the optimal behaviour of the algorithm should be and to deduce from that a fair estimate of some of the parameters.

Parameter Estimation.

- The parameters r_2 and r_1 are meant to compensate multiple representations of the intermediate solutions. When the algorithm performs optimally, we should have

  $$S_1 \approx \mu_2 C_2 \approx \frac{\binom{k+\ell}{p_1}}{2^{r_1+r_2}} \text{ and } S_0 \approx \mu_1 C_1 \approx \frac{\binom{k+\ell}{p}}{2^{\ell}},$$

  that is

  $$2^{r_2} \approx \frac{\mu_2 \binom{(k+\ell)/2}{p_2/2}^4}{\binom{k+\ell}{p_1}} \text{ and } 2^{r_1+r_2} \approx \frac{\mu_1 \binom{k+\ell}{p_1}^2}{\binom{k+\ell}{p}}. \qquad (6)$$

  Next, assuming that last two level of the algorithm are dominant (which is true in practice, at least when we are close to the optimal parameters), the workfactor will be proportional to $C_2 + C_1$ and is likely to be minimal when $C_1 \approx C_2$, that is

  $$2^{\ell} \approx \mu_2 \binom{k+\ell}{p_1}. \qquad (7)$$

  Between (6) and (7) we have three equation which provide values of ℓ, r_1, and r_2 for any fixed values of p, e_1, and e_2.
- Assuming now that ℓ, r_1, and r_2 verify (6)-(7) and that the algorithm cost is dominated by the last two steps, the workfactor will be proportional to

  $$\frac{C_1}{\mathcal{P}(p,\ell)} = \frac{1}{\varepsilon(p,\ell)} \frac{1}{\mu_2 \mu_1 \binom{k+\ell}{p_1}}.$$

  In this expression, only μ_2 and ℓ vary with e_2. We will assume the variations of ℓ with e_2 are negligible. In that case, the value of e_2 for which the expression is minimal is such that μ_2 is maximal, that is

  $$e_2 \approx \frac{p_2^2}{k+\ell} \approx \frac{1}{16} \frac{(p+2e_1)^2}{k+\ell}.$$

  Now if we fix everything but e_1, the best value for e_1 will be such that $\mu_1 \mu_2 \binom{k+\ell}{p_1}$ is maximal. Experimentally we find $e_1 \approx p/4$.

The approach is heuristic, but for any fixed value of p we look for the optimal parameters in the vicinity of the above estimate for $(\ell, r_1, r_2, e_1, e_2)$. This works very well in practice. Experimentally, the convexity assumption doesn't seem to hold for all parameters and so the search is a bit more complex than for the other variants.

4 Experimental Results

We implemented our estimates and give numbers for various code parameters corresponding to typical instances of McEliece encryption scheme with Goppa codes or MDPC codes [19]. The table gives two workfactors for each algorithm. The leftmost is computed with unconstrained parameters: the parameters can take any integer value, even those not allowed by the algorithm. For instance odd values of p do not seem to fit but could make sense with a clever implementation. The rightmost value corresponds to more realistic constraints: p even for SD-ISD, p multiple of 4 for MMT-ISD, and p, p_1, p_2 even for BJMM-ISD. It is interesting to note that the speedup for BJMM-ISD, though small, is already measurable for cryptographic sizes.

Table 5. Unconstrained and constrained workfactors for ISD (in column operations)

(n, k, t)	SD-ISD		MMT-ISD		BJMM-ISD	
$(1024, 524, 50)$	55.42	55.60	54.29	54.75	52.50	52.90
$(2048, 1696, 32)$	81.60	81.60	79.32	79.50	75.78	76.82
$(4096, 3844, 21)$	81.23	81.23	78.11	78.88	74.34	78.46
$(4096, 3616, 40)$	121.38	121.38	118.048	119.08	114.62	118.90
$(8192, 7945, 19)$	89.54	89.54	85.87	87.16	82.58	87.43
$(8192, 7815, 29)$	122.66	122.66	118.67	120.31	115.65	120.84
$(9600, 4800, 84)$	88.73	88.73	87.75	87.75	85.82	86.16
$(22272, 14848, 85)$	138.06	138.06	137.07	137.07	134.78	135.35

References

1. D. Augot, M. Finiasz, P. Gaborit, S. Manuel, and N. Sendrier. SHA-3 proposal: FSB. Submission to the SHA-3 NIST competition, 2008.
2. D. Augot, M. Finiasz, and N. Sendrier. A fast provably secure cryptographic hash function. Cryptology ePrint Archive, Report 2003/230, 2003. http://eprint.iacr.org/.
3. Gregory Bard and Martin Albrecht. M4ri(e)- linear algebra over $\mathbf{F}_2$ (and $\mathbf{F}_{2^e}$). Free Open Source Software. `http://m4ri.sagemath.org/index.html`.
4. Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in $2^{n/20}$: How 1+1=0 improves information set decoding. In D. Pointcheval and T. Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *LNCS*, pages 520–536. Springer, 2012.
5. D.J. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. In J. Buchmann and J. Ding, editors, *Post-Quantum Cryptography*, volume 5299 of *LNCS*, pages 31–46. Springer, 2008.
6. D.J. Bernstein, T. Lange, and C. Peters. Smaller decoding exponents: Ball-collision decoding. In P. Rogaway, editor, *Advances in Cryptology - CRYPTO 2011*, volume 6841 of *LNCS*, pages 743–760. Springer, 2011.

7. P. Camion and J. Patarin. The knapsack hash function proposed at CRYPTO '89 can be broken. In D.W. Davies, editor, *Advances in Cryptology - EUROCRYPT '91*, volume 547 of *LNCS*, pages 39–53. Springer, 1991.
8. A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, January 1998.
9. J.-S. Coron and A. Joux. Cryptanalysis of a provably secure cryptographic hash function. Cryptology ePrint Archive, Report 2004/013, 2004. http://eprint.iacr.org/.
10. N. Courtois, M. Finiasz, and N. Sendrier. How to achieve a McEliece-based digital signature scheme. In C. Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 157–174. Springer, 2001.
11. I. Dumer. On minimum distance decoding of linear codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, pages 50–52, Moscow, 1991.
12. M. Finiasz and N. Sendrier. Security bounds for the design of code-based cryptosystems. In Mitsuru Matsui, editor, *Advances in Cryptology - ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 88–105. Springer, 2009.
13. J.-B. Fischer and J. Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In Ueli Maurer, editor, *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *LNCS*, pages 245–255. Springer, 1996.
14. P. Gaborit and M. Girault. Lightweight code-based identification and signature. In *IEEE Conference, ISIT 2007*, pages 191–195, Nice, France, July 2007. IEEE.
15. P. Gaborit, C. Laudaroux, and N. Sendrier. Synd: a very fast code-based stream cipher with a security reduction. In *IEEE Conference, ISIT 2007*, pages 186–190, Nice, France, July 2007. IEEE.
16. N. Howgrave-Graham and A. Joux. New generic algorithms for hard knapsacks. In H. Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 235–256. Springer, 2010.
17. A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In D.H. Lee and X. Wang, editors, *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *LNCS*, pages 107–124. Springer, 2011.
18. R.J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Prog. Rep., Jet Prop. Lab., California Inst. Technol., Pasadena, CA*, pages 114–116, January 1978.
19. Rafael Misoczki, Jean-Pierre Tillich, Nicolas Sendrier, and Paulo S. L. M. Barreto. Mdpc-mceliece: New mceliece variants from moderate density parity-check codes. Cryptology ePrint Archive, Report 2012/409, 2012. http://eprint.iacr.org/.
20. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Prob. Contr. Inform. Theory*, 15(2):157–166, 1986.
21. J.K. Omura. Iterative decoding of linear codes by a modulo-2 linear programm. *Discrete Mathematics*, 3:193–208, 1972.
22. R. Overbeck and N. Sendrier. Code-based cryptography. In D.J. Bernstein, J. Buchmann, and E. Dahmen, editors, *Post-Quantum Cryptography*, pages 95–145. Springer, 2009.
23. C. Peters. *Curves, Codes, and Cryptography.* PhD thesis, Technische Universiteit Eindhoven, 2011.
24. E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions*, IT-8:S5–S9, 1962.

25. M.-J. Saarinen. Linearization attacks against syndrome based hashes. In K. Srinathan, C. Pandu Rangan, and M. Yung, editors, *Indocrypt 2007*, volume 4859 of *LNCS*, pages 1–9. Springer, 2007.
26. N. Sendrier. Decoding one out of many. Cryptology ePrint Archive, Report 2011/367, 2011. http://eprint.iacr.org/.
27. J. Stern. A method for finding codewords of small weight. In G. Cohen and J. Wolfmann, editors, *Coding theory and applications*, volume 388 of *LNCS*, pages 106–113. Springer, 1989.
28. J. Stern. A new identification scheme based on syndrome decoding. In D.R. Stinson, editor, *Advances in Cryptology - CRYPTO '93*, volume 773 of *LNCS*, pages 13–21. Springer, 1993.
29. P. Véron. Improved identification schemes based on error-correcting codes. *AAECC*, 8(1):57–69, January 1997.
30. D. Wagner. A generalized birthday problem. In M. Yung, editor, *Advances in Cryptology - CRYPTO 2002*, volume 2442 of *LNCS*, pages 288–303. Springer, 2002.