Decoding One Out of Many

Nicolas Sendrier

INRIA Paris-Rocquencourt, Project-Team SECRET
nicolas.sendrier@inria.fr

Abstract. Generic decoding of linear codes is the best known attack against most code-based cryptosystems. Understanding and measuring the complexity of the best decoding technique is thus necessary to select secure parameters. We consider here the possibility that an attacker has access to many cryptograms and is satisfied by decrypting (i.e. decoding) only one of them. We show that, in many cases of interest in cryptology, a variant of Stern's collision decoding can be adapted to gain a factor almost $\sqrt{N}$ when N instances are given. If the attacker has access to an unlimited number of instances, we show that the attack complexity is significantly lower, in fact raised by a power slightly larger than $2/3$. Finally we give indications on how to counter those attacks.

1 Introduction

Code-based cryptography have attracted a lot of interest in the past few years, accompanying the rise of post-quantum cryptography. It allows public-key encryption scheme [McE78,Nie86], zero-knowledge protocols [Ste93,Vér97,GG07], digital signature [CFS01], hash functions [AFG$^+$08,BLPS11], stream ciphers [FS96,GLS07] to mention only the most classical primitives. The common point of all code-based cryptographic primitives is the fact that they rely on the hardness of decoding a linear code with no apparent algebraic structure. The problem is NP-hard [BMvT78], and in fact, the parameter selection for those systems is based on the best knows decoding techniques, usually the collision decoding [Ste89] and its variants, and sometimes the generalized birthday algorithm (GBA) [CP91,Wag02].

The complexity of information set decoding (ISD) for cryptographic application has been widely studied and is well understood (see [Pet11] for today's state of the art). In this work, we consider the case where the attacker knows many instances of the problem and wishes to solve only one of them. Doing this with ISD has already been done [JJ02], but the analysis does not help us to understand how much we can gain. With GBA, there an unpublished attack, attributed to Bleichenbacher, which saves, more or less, a factor $\sqrt{N}$ when N instances are decoded simultaneously (and the solution for only one is needed).

When saving a factor $\sqrt{N}$ to an attack, there is necessarily a limit to N. If the initial cost was T, we must have $T/\sqrt{N} \leq N$ because each instance must be read once. Thus the optimal value for N is $T^{2/3}$ which is also the new cost for the attack.

We will show in this paper that, when the number of errors to decode is smaller than the Gilbert-Varshamov distance, collision decoding can be adapted to save a factor $N^{0.5-c}$ (with a small positive c for which we have an estimate) when decoding one out of N instances. Also, if the number of instances is unlimited, we show that the cost of the decoding is raised to the power $2/3 + c'$ (with a small positive c' for which we have an estimate).

We will first analyze an abstract variant of ISD, similar to the one of [FS09]. We will then show how this algorithm and its analysis can be extended to the case of many instance and provide some estimates of what this modified algorithm can gain.

This new attack constitute a threat which must be considered. We briefly explain in the conclusion how to completely avoid this threat. The countermeasure is simple but it is a new feature to consider when implementing code-based cryptography.

Notation:

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

2 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 $eH^T = s$.*

We will denote $\mathrm{CSD}(H, s, w)$ the above problem and the set of its solutions. Decoding is one the the prominent algorithmic problem 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 [McE78,Nie86] and digital signature [CFS01], zero-knowledge protocols based on codes [Ste93,Vér97,GG07], hash-function [AFG$^+$08], PRNG and stream ciphers [FS96,GLS07] and many others, decoding is the most threatening attack and therefore is a key point in the parameter selection.

2.1 Generic Decoding Algorithms

There are two main techniques for addressing CSD in cryptology. The most ancient one, Information Set Decoding (ISD) can be traced back to Prange [Pra62].

The variants useful today in cryptology all derive more or less from Stern's algorithm [Ste89], which we will call collision decoding, following [Pet11]. It was implemented (with various improvements) in [CC98] then in [BLP08] which reports the first successful attack on the original parameter set. General lower bounds were proposed [FS09]. The last published variant is ball-collision decoding [BLP11] which features a better decoding exponent than collision decoding.

ISD is the best know technique when the error weight w is smaller than the Gilbert-Varshamov distance, which is defined as the smallest integer d_0 such that

$$\binom{n}{d_0} \geq 2^r.$$

When w is larger than d_0 the Generalized Birthday Algorithm (GBA) [Wag02] (order 2 GBA was previously published in [CP91]) is sometimes more efficient than ISD. The first use of GBA for decoding was proposed in [CJ04] for attacking an early version of FSB [AFG$^+$08]. Finally, when w gets even larger (typically $w > r/4$) the best known technique is linearization [Saa07].

2.2 Decoding One Out of Many Instances

In this work we will also consider another scenario where the attacker dispose of a large number of instances (H, s, w) where the parity check matrix H and the error weight w are identical, but the syndrome s runs over some large set.

Problem 2 (Computational Syndrome Decoding - Multi) *Given a matrix $H \in \{0,1\}^{r \times n}$, a set $\mathcal{S} \subset \{0,1\}^r$, and an integer $w > 0$, find a word $e \in \{0,1\}^n$ of Hamming weight $\leq w$ such that $eH^T \in \mathcal{S}$.*

For convenience, we will also denote $\mathrm{CSD}(H, \mathcal{S}, w)$ this problem and the set of its solutions. It has been addressed already using GBA by Bleichenbacher (unpublished, reported in [OS09]) for attacking the digital signature CFS. In practice, the attacker builds a large number N of favorable instances of the decoding problem and gain a speedup of $\sqrt{N}$. This reduces the order of magnitude of the attack from $O(2^{r/2})$ to $O(2^{r/3})$. A variant of CFS resistant to this attack was recently published [Fin10].

An attempt at using ISD with multiple instances was already made in [JJ02]. We revisit here that work in a more general setting and with a more thorough complexity analysis.

3 A Generalized Information Set Decoding Algorithm

Following other works [LB88,Leo88], J. Stern describes in [Ste89] an algorithm to find a word of weight w in a binary linear code of length n and dimension k (and codimension $r = n - k$). The algorithm uses two additional parameters p and ℓ (both positive integers). We present here a generalized version, similar to

For any fixed values of n, r and w, the following algorithm uses four parameters: two integers $p > 0$ and $\ell > 0$ and two sets $W_1 \subset \mathcal{S}_{k+\ell}(\mathbf{0}, p_1)$ and $W_2 \subset \mathcal{S}_{k+\ell}(\mathbf{0}, p_2)$ where p_1 and p_2 are positive integers such that $p_1 + p_2 = p$.

procedure main_isd
input: $H_0 \in \{0,1\}^{r\times n}$, $s_0 \in \{0,1\}^r$
    `repeat`
(ISD 0) $\left\{\begin{array}{l} P \leftarrow \text{random } n \times n \text{ permutation matrix} \\ (H', H'', U) \leftarrow \text{PartialGaussElim}(H_0 P) \qquad \text{// as in (1)} \\ s \leftarrow s_0 U^T \end{array}\right.$
        $e \leftarrow \text{isd\_loop}(H', H'', s)$
    `while` $e = \text{FAIL}$
    `return` (P, e)

procedure isd_loop
input: $H' \in \{0,1\}^{\ell\times(k+\ell)}$, $H'' \in \{0,1\}^{(r-\ell)\times(k+\ell)}$, $s \in \{0,1\}^r$
    `for all` $e_1 \in W_1$
(ISD 1) $\left\{\begin{array}{l} i \leftarrow e_1 H'^T,\ s''_1 \leftarrow e_1 H''^T \\ \text{write}(e_1, s''_1, i) \qquad \text{// stores } (e_1, s''_1) \text{ at index } i \end{array}\right.$
    `for all` $e_2 \in W_2$
(ISD 2) $\left\{\begin{array}{l} i \leftarrow s' + e_2 H'^T,\ s''_2 \leftarrow s'' + e_2 H''^T \\ \text{Elts} \leftarrow \text{read}(i) \qquad \text{// extracts the elements stored at index } i \end{array}\right.$
        `for all` $(e_1, s''_1) \in \text{Elts}$
(ISD 3) $\left\{\begin{array}{l} \texttt{if } \text{wt}\,(s''_1 + s''_2) = w - p \\ \qquad \texttt{return } e_1 + e_2 \qquad \text{(SUCCESS)} \end{array}\right.$
    `return` FAIL (FAIL)

Algorithm 1. Generalized ISD algorithm

the one presented in [FS09], which acts on the parity check matrix H_0 of the code (instead of the generator matrix). Table 1 describes the algorithm. The partial Gaussian elimination of $H_0 P$ consists in finding U ($r \times r$ and non-singular) and H (and H', H'') such that[1]

$$
UH_0P = H = \begin{array}{c} \\ \\ \ell \end{array}
\begin{array}{c|c} \multicolumn{1}{c}{r-\ell} & \multicolumn{1}{c}{k+\ell} \\ \hline \begin{matrix} 1 & & \\ & \ddots & \\ & & 1 \end{matrix} & H'' \\ \hline 0 & H' \end{array},
\quad s^T = U s_0^T = \begin{bmatrix} s''^T \\ \hline s'^T \end{bmatrix} \tag{1}
$$

where U is a non-singular $r \times r$ matrix. We have $e \in \text{CSD}(H, s, w)$ if and only if $eP^T \in \text{CSD}(H_0, s_0, w)$. Let (P, e') be the output of the algorithm and $e'' = s'' + e'H''^T$ the word $e = (e'', e')$ is in $\text{CSD}(H, s, w)$.

[1] in the unlikely event that the first $r - \ell$ columns are dependent, we change P

Definition 1. *For any fixed value of n, r and w, we denote* $\mathrm{WF}_{\mathrm{ISD}}(n, r, w)$ *the minimal work factor (average cost in elementary operations) of Algorithm 1 to produce a solution to* CSD *(provided there is a solution), for any choices of parameters ℓ, p, W_1 and W_2.*

In the literature, elementary operations are often binary instructions. Our purpose here is to obtain a measure allowing us to compare algorithms, to measure the impact of multiple instances and to make an asymptotic analysis. Any reasonably fixed polynomial time (in n) "elementary operation" will serve that purpose.

3.1 Links With the Other Variants of Collision Decoding

Information set decoding is an old decoding technique [Pra62], the variants of interest today for cryptanalysis derive from Stern's collision decoding [Ste89]. The algorithm we present here is closer to the "Punctured Split Syndrome Decoding" of Dumer [Dum91,Bar98]. Depending on how the sets W_1 and W_2 are chosen, we may obtain any known variant, including the recent ball-collision decoding [BLP11]. Of course the Algorithm 1 is an abstraction. An effective algorithm, not to speak of its implementation must include a description of how the parameters p and ℓ are chosen (something we will do) and how the sets W_1 and W_2 are selected (something we will not do completely). Our main purpose in this work is to estimate the impact of having multiple instances. This requires some flexibility in the choice of the sizes of W_1 and W_2 which is relatively natural in our abstract model, but not straightforward, though probably possible, in the above mentioned variants. We believe that the evolution of the complexity given in (9) and (10) between the single and multiple instances scenarios can be obtained for most variants of collision decoding after proper adjustments.

4 Cost Estimation

We will neglect all control instructions and assume that counting only the instructions in blocks (ISD i) will give an accurate estimation of the algorithm cost. For $i = 0, 1, 2, 3$ we will denote K_i the average cost in elementary operations (whatever that means) for executing the block of instructions (ISD i).

Given n, r, w, p and ℓ, we will denote

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

the probability for some $e' \in \mathcal{S}_{k+\ell}(\mathbf{0}, p)$ to be a valid output of a particular execution of isd_loop given that the input (H_0, s_0) of Algorithm 1 has a solution. Given, in addition, W_1 and W_2 we denote

$$\mathcal{P}(p, \ell) = 1 - (1 - \varepsilon(p, \ell))^{|W_1+W_2|} \tag{3}$$

where $W_1+W_2 = \{e_1+e_2 \mid (e_1,e_2) \in W_1 \times W_2\}$, the probability of one particular execution of isd_loop to return and element of $\mathcal{S}_{k+\ell}(\mathbf{0},p)$ given that the input (H_0,s_0) of Algorithm 1 has a solution. In addition to the usual random coding assumption (pseudo-randomness of syndromes) which have already been used in the above probabilities, we will admit the following.

Assumptions and approximations:

1. K_0, K_1, K_2, and K_3 are independent of p, ℓ, W_1 and W_2.
2. All sums e_1+e_2 for $(e_1,e_2) \in W_1 \times W_2$ are distinct and $|W_1||W_2| \leq \binom{k+\ell}{p}$.
3. Up to a (small) constant factor we have for any $x \ll 1$ and any integer N

$$1 - (1-x)^N \approx \min(1, xN)$$

Those assumptions and approximations will not cost more than a small constant factor on the cost estimations we will compute later in this paper.

Proposition 3 *For an input (H_0,s_0) such that* $\mathrm{CSD}(H_o,s_0,w) \neq \emptyset$*, the Algorithm 1 will stop after executing*

$$\mathcal{T}(p,\ell) \approx \frac{K_0}{\mathcal{P}(p,\ell)} + \frac{K_1|W_1|}{\mathcal{P}(p,\ell)} + \frac{K_2}{|W_1|\varepsilon(p,\ell)} + \frac{K_3}{2^\ell \varepsilon(p,\ell)} \tag{4}$$

elementary operations in average.

Proof: The two leftmost terms are straightforward as the average number of calls to isd_loop is equal $1/\mathcal{P}(p,\ell)$. One particular execution of (ISD 2) will inspect $|W_1|$ different sums e_1+e_2 and thus succeeds with probability

$$\pi_2 = 1 - (1-\varepsilon(p,\ell))^{|W_1|}.$$

When the parameters are optimal we have $\varepsilon(p,\ell)|W_1| \ll 1$ and thus $\pi_2 \approx \varepsilon(p,\ell)|W_1|$ which accounts for the third term in (4). Finally, if the call to isd_loop fails, the block (ISD 3) will be called in average $|W_1||W_2|/2^\ell$ times. Thus if π_3 is its probability of success, we have

$$1 - \mathcal{P}(p,\ell) = (1-\pi_3)^{\frac{|W_1||W_2|}{2^\ell}} \text{ and thus } \pi_3 = 1 - (1-\varepsilon(p,\ell))^{2^\ell}.$$

As $\varepsilon(p,\ell)2^\ell \ll 1$, we have $\pi_3 = \varepsilon(p,\ell)2^\ell$ and thus the rightmost term of (4). □

An easy consequence of this proposition is that the minimal cost for Algorithm 1 is obtained when W_2 has maximal size (everything else being fixed), that is, within our assumptions, when $|W_1||W_2| = \binom{k+\ell}{p}$. At this point, $\mathcal{P}(p,\ell)$ is independent of W_1 and the complexity is minimal when the two middle terms of (4) equals, that is when

$$|W_1| = \mathcal{L}(p,\ell) = \sqrt{\frac{K_2\mathcal{P}(p,\ell)}{K_1\varepsilon(p,\ell)}} = \sqrt{\frac{K_2}{K_1}} \min\left(\sqrt{\frac{1}{\varepsilon(p,\ell)}}, \sqrt{\binom{k+\ell}{p}}\right) \tag{5}$$

which is consistent with the results of [FS09]. We have

$$\mathrm{WF}_{\mathrm{ISD}}(n,r,w) = \min_{p,\ell} \mathcal{T}(p,\ell)$$

where

$$\mathcal{T}(p,\ell) \approx \frac{K_0}{\mathcal{P}(p,\ell)} + \frac{2K_2}{\mathcal{L}(p,\ell)\varepsilon(p,\ell)} + \frac{K_3}{2^\ell \varepsilon(p,\ell)}. \tag{6}$$

Note that when $\varepsilon(p,\ell)\binom{k+\ell}{p} < 1$, the "min" in (5) is obtained for rightmost term and W_1 and W_2 have (approximatively) the same size. Else $\mathcal{P}(p,\ell) = 1$ (which happens only when w is large) and the optimal choice consists in choosing W_1 smaller than W_2.

4.1 Lower Bound

Assuming that $K_0 = 0$ (we neglect the Gaussian elimination), the cost estimate becomes

$$\mathcal{T}(p,\ell) \approx \frac{2K_2}{\mathcal{L}(p,\ell)\varepsilon(p,\ell)} + \frac{K_3}{2^\ell \varepsilon(p,\ell)} \tag{7}$$

and because the first term is increasing and the second is decreasing (for parameters of cryptologic interest) we have for all p

$$\frac{1}{2}\mathcal{T}(p,\ell_1) \le \min_{\ell} \mathcal{T}(p,\ell) \le \mathcal{T}(p,\ell_1)$$

where $\ell_1(p)$, or ℓ_1 for short, is the unique integer in $[0, r[$ such that the two terms in $\mathcal{T}(p,\ell)$ are equal, that is

$$\ell_1 = \log_2\left(\frac{K_3}{2K_2}\mathcal{L}(p,\ell_1)\right) = \log_2\left(\frac{K_3}{2\sqrt{K_1 K_2}}\sqrt{\frac{\mathcal{P}(p,\ell_1)}{\varepsilon(p,\ell_1)}}\right).$$

The lower bound is

$$\mathrm{WF}_{\mathrm{ISD}}(n,r,w) \ge \min_{p} \frac{1}{2}\mathcal{T}(p,\ell_1)$$

and the various forms of $\mathcal{T}(p,\ell_1)$ give various interpretations of the complexity

$$\mathcal{T}(p,\ell_1) = \frac{2K_1\mathcal{L}(p,\ell_1)}{\mathcal{P}(p,\ell_1)} = \frac{2K_3}{2^{\ell_1}\varepsilon(p,\ell_1)} = \frac{2K_2}{\mathcal{L}(p,\ell)\varepsilon(p,\ell_1)} = \frac{2\sqrt{K_1K_2}}{\sqrt{\mathcal{P}(p,\ell)\varepsilon(p,\ell_1)}}$$

This bound is tight if the Gaussian elimination cost is negligible (which is often the case in practice, see Table 2).

4.2 Some Numbers

For Table 3 will assume that $K_0 = nr$, $K_1 = K_2 = 1$, and $K_3 = 2$. The elementary operation being a "column operation": a column addition or the computation of a Hamming weight, possibly accompanied by a memory access.

(n, r, w)	$\log_2(\mathrm{WF}_{\mathrm{ISD}})$	$\min_p \log_2 \frac{\mathcal{T}(p, \ell_1)}{2}$
$(2048, 352, 32)$	81.0	80.5
$(2048, 781, 71)$	100.7	100.1
$(4096, 252, 21)$	80.4	80.0
$(4096, 540, 45)$	128.3	127.9
$(8192, 416, 32)$	128.8	128.4
$(2^{16}, 144, 11)$	70.2	70.1
$(2^{16}, 160, 12)$	79.4	79.3
$(2^{18}, 162, 11)$	78.9	78.8
$(2^{20}, 180, 11)$	87.8	87.7
$(5 \cdot 2^{18}, 640, 160)$	91.8	90.9
$(7 \cdot 2^{18}, 896, 224)$	126.6	125.7
$(2^{21}, 1024, 256)$	144.0	143.1
$(23 \cdot 2^{16}, 1472, 368)$	205.9	205.0
$(31 \cdot 2^{16}, 1984, 496)$	275.4	274.6

Table 2. Workfactor estimates and lower bounds for generalized ISD. The code parameters of the first block of numbers corresponds to encryption, the second to the CFS digital signature scheme and the third to collision search in the (non-regular) FSB hash function

The cost for (ISD 1) and (ISD 2) can be reduced to 1 by "reusing additions", as explained in [BLP08]. The "column" has size r bits ($r - \ell$ for (ISD 3)), however we need in practice ℓ bits for computing the index in (ISD 1) and (ISD 2), and for (ISD 3) we only need in average $2(w-p)$ additional bits for deciding whether or not we reach the target weight. This sets the "practical column size" to $\ell + 2(w-p)$ instead of r. We claim that up to a small constant factor, this measure will give a realistic account for the cost of a software implementation.

4.3 Variations With the Parameter p

There exists an expression for the optimal, or nearly optimal value $\ell_1(p)$ of ℓ for a given n, r, w, and p. Even though it defines $\ell_1(p)$ implicitly, it gives an intuition of the significance and variations of ℓ_1. Finding something similar for p given n, r, and w (with $\ell = \ell_1(p)$ of course) seems to be more challenging. However, we observe that, when w is much smaller than the Gilbert-Varshamov distance (typically for encryption), the value of $\mathcal{T}(p, \ell_1(p))$ varies relatively slowly with p when p is close to the optimal.

As an illustration, we give in Table 3 values of $\mathcal{T}(p, \ell)$ (computed with (6)) for various optimal pairs (p, ℓ) and code parameters.

5 Decoding One Out of Many

We assume now that we have to solve $\mathrm{CSD}(H, \mathcal{S}, w)$ for set of $\mathcal{S}$ of N independent syndromes which all have a solution. We describe a procedure for that in

$(n,r,w)=(4096,540,45)$												
p	6	7	8	9	10	11	**12**	13	14	15	16	17
ℓ	34	38	43	47	51	56	**60**	64	68	72	76	80
$\log_2 \mathcal{T}(p,\ell)$	129.4	129.0	128.7	128.5	128.4	128.3	**128.3**	128.4	128.6	128.9	129.2	129.6

$(n,r,w)=(2^{20},180,11)$							
p	4	5	6	7	8	9	**10**
ℓ	41	50	59	68	77	86	**94**
$\log_2 \mathcal{T}(p,\ell)$	106.1	102.1	98.2	94.6	91.2	88.1	**87.7**

$(n,r,w)=(2^{21},1024,256)$												
p	11	12	13	14	15	**16**	17	18	19	20	21	22
ℓ	103	112	121	129	138	**144**	145	146	147	148	148	149
$\log_2 \mathcal{T}(p,\ell)$	158.4	155.1	151.8	148.5	145.3	**144.0**	144.9	145.8	146.7	147.7	148.6	149.5

Table 3. Cost estimate for various optimal (p,ℓ) the first (top) table corresponds to encryption, the second to digital signature and the third to hashing

Algorithm 4. We keep the same notations and use the same assumptions and approximations as in §4. We denote

$$\mathcal{P}_N(p,\ell) = 1 - (1-\varepsilon(p,\ell))^{N|W_1||W_2|} \approx \min\left(1, \varepsilon(p,\ell)N|W_1||W_2|\right)$$

the probability for one execution of doom_loop to succeed. We have a statement very similar to Proposition 3.

Proposition 4 *For an input $(H_0,\mathcal{S}_0)$ such that* $\mathrm{CSD}(H_o,s_0,w)\neq\emptyset$ *for all $s_0\in\mathcal{S}_0$ the Algorithm 4 will stop after executing*

$$\mathcal{T}_N(p,\ell) \approx \frac{K_0}{\mathcal{P}_N(p,\ell)} + \frac{K_1|W_1|}{\mathcal{P}_N(p,\ell)} + \frac{K_2}{|W_1|\varepsilon(p,\ell)} + \frac{K_3}{2^\ell\varepsilon(p,\ell)} \tag{8}$$

elementary operations in average.

We omit the proof which is similar to the proof of Proposition 3 with an identical expression for the complexity except for $\mathcal{P}_N(p,\ell)$ (which grows with N).

5.1 Cost of Linear Algebra

The constant K_0 will include, in addition to the Gaussian elimination, the computation of all the s_oU^T for $s_0\in\mathcal{S}_0$. This multiplies the cost, at most, by a factor $N=|\mathcal{S}_0|$. On the other hand, as long as $N\leq 1/\varepsilon(p,\ell)\binom{k+\ell}{p}$ (taking larger N does not make sense) the probability $\mathcal{P}_N(p,\ell)$ is N times larger than before and thus the ratio $K_0/\mathcal{P}_N(p,\ell)$ do not increase. The total cost $\mathcal{T}_N(p,\ell)$ is smaller than $\mathcal{T}(p,\ell)$, so the relative contribution of the linear algebra will increase, but the simplification $K_0=0$ remains reasonable as long as $\mathcal{P}_N(p,\ell)\ll 1$.

When N is close or equal to $1/\varepsilon(p,\ell)\binom{k+\ell}{p}$, as in §5.3, the situation is not so simple. With fast binary linear algebra computing all the s_oU^T will require about

For any fixed values of n, r and w, the following algorithm uses four parameters: two integers $p > 0$ and $\ell > 0$ and two sets $W_1 \subset \mathcal{S}_{k+\ell}(\mathbf{0}, p_1)$ and $W_2 \subset \mathcal{S}_{k+\ell}(\mathbf{0}, p_2)$ where p_1 and p_2 are positive integers such that $p_1 + p_2 = p$.

procedure main_doom
input: $H_0 \in \{0,1\}^{r\times n}$, $\mathcal{S}_0 \subset \{0,1\}^r$
`repeat`
(DOOM 0) $\begin{cases} P \leftarrow \text{random } n \times n \text{ permutation matrix} \\ (H', H'', U) \leftarrow \text{PartialGaussElim}(H_0 P) & \text{// as in (1)} \\ \mathcal{S} \leftarrow \{s_0 U^T \mid s_0 \in \mathcal{S}_0\} \end{cases}$
$e \leftarrow \text{doom\_loop}(H', H'', \mathcal{S})$
`while` $e = $ FAIL
`return` (P, e)

procedure doom_loop
input: $H' \in \{0,1\}^{\ell\times(k+\ell)}$, $H'' \in \{0,1\}^{(r-\ell)\times(k+\ell)}$, $\mathcal{S} \subset \{0,1\}^r$
`for all` $e_1 \in W_1$
(DOOM 1) $\begin{cases} i \leftarrow e_1 H'^T,\ s_1'' \leftarrow e_1 H''^T \\ \text{write}(e_1, s_1'', i) & \text{// stores } (e_1, s_1'') \text{ at index } i \end{cases}$
`for all` $e_2 \in W_2$
`for all` $s = (s', s'') \in \mathcal{S}$
(DOOM 2) $\begin{cases} i \leftarrow s' + e_2 H'^T,\ s_2'' \leftarrow s'' + e_2 H''^T \\ \text{Elts} \leftarrow \text{read}(i) & \text{// extracts the elements stored at index } i \end{cases}$
`for all` $(e_1, s_1'') \in \text{Elts}$
(DOOM 3) $\begin{cases} \texttt{if } \text{wt}(s_1'' + s_2'') = w - p \\ \quad \texttt{return } e_1 + e_2 & \text{(SUCCESS)} \end{cases}$
`return` FAIL (FAIL)

Algorithm 4. DOOM ISD algorithm

$Nr/\log_2 N$ column operations. For the extremal values of N of §5.3 (the worst case), assuming $K_1 = K_2 = K_3/2 = 1$, we have $\mathcal{P}_n(p, \ell) = 1$ and a complexity

$$\approx \frac{Nr}{\log_2 N} + 2^{\ell+2} \text{ with } N = \frac{2^{2\ell}}{\binom{k+\ell}{p}} \leq 2^\ell$$

Unless we precisely use the optimal value of p, for which $N \approx \binom{k+\ell}{p} \approx 2^\ell$, the ratio $N/2^\ell$ will be significantly smaller than 1 and $K_0 = 0$ provides an accurate estimate. Finally when p is optimal (this value, by the way, is not necessarily an integer) we have a complexity of the form $2^\ell(r/\ell + 4)$ and we cannot completely neglect r/ℓ compared with 4. For the sake of simplicity, we do it nevertheless.

5.2 Complexity Gain From Multiple Instances

We will denote

$$\text{WF}_{\text{ISD}}^{(N)}(n, r, w) = \min_{p,\ell} \mathcal{T}_N(p, \ell)$$

and the gain we wish to estimate is the ratio

$$\gamma = \log_N \frac{\mathrm{WF}_{\mathrm{ISD}}(n,r,w)}{\mathrm{WF}^{(N)}_{\mathrm{ISD}}(n,r,w)}$$

which we expect to be close to 1/2. First, we must have

$$N \leq \frac{1}{\varepsilon(p,\ell)\binom{k+\ell}{p}} = \frac{\min\left(2^r, \binom{n}{w}\right)}{\binom{r-\ell}{w-p}\binom{k+\ell}{p}}$$

else there is nothing to gain. Within this bound, we have

$$\mathcal{P}_N(p,\ell) = N\varepsilon(p,\ell)\binom{k+\ell}{p} \text{ and } \mathcal{L}_N(p,\ell) = \sqrt{\tfrac{K_2}{K_1}}\sqrt{N\binom{k+\ell}{p}}$$

and (assuming $K_0 = 0$)

$$\mathcal{T}_N(p,\ell) = \frac{2\sqrt{K_1K_2}}{\sqrt{N\binom{k+\ell}{p}}\varepsilon(p,\ell)} + \frac{K_3}{2^\ell \varepsilon(p,\ell)}.$$

The same analysis as in §4.1 will tell us that the above sum is minimal (up to a factor at most two) when its two terms are equal, that is when $\ell = \ell_N(p)$, or ℓ_N for short, where

$$\ell_N = \log_2\left(\frac{K_3\sqrt{N\binom{k+\ell_N}{p}}}{2\sqrt{K_1K_2}}\right).$$

Proposition 5 *For a given p, we have*

$$\log_N \frac{\mathcal{T}(p,\ell_1)}{\mathcal{T}_N(p,\ell_N)} \approx \frac{1}{2} - c(p) \text{ where } c(p) = \frac{1}{2\ln 2}\frac{w-p}{r-\ell_1-\frac{w-p-1}{2}}.$$

Proof: We have

$$\ell_N = \log_2\left(\frac{K_3\sqrt{N\binom{k+\ell_N}{p}}}{2\sqrt{K_1K_2}}\right) \text{ and } \ell_1 = \log_2\left(\frac{K_3\sqrt{\binom{k+\ell_1}{p}}}{2\sqrt{K_1K_2}}\right)$$

and if we consider only the first order variations, we have $\ell_N \approx \ell_1 + \frac{1}{2}\log_2 N$. Because we have

$$\frac{d}{da}\binom{a}{b} = \binom{a}{b}\Delta(a,b) \text{ where } \Delta(a,b) = \sum_{i=0}^{b-1}\frac{1}{a-i} \approx \frac{b}{a-\frac{b-1}{2}}$$

it follows that, keeping only the first order variations, we have

$$\varepsilon(p,\ell_N) \approx \varepsilon(p,\ell_1)\exp(-c(p)\log N)$$

where $c(p) \approx \Delta(r-\ell_1, w-p)/2\ln(2)$. Finally

$$\frac{\mathcal{T}(p,\ell_1)}{\mathcal{T}_N(p,\ell_N)} = \frac{2^{\ell_N}\varepsilon(p,\ell_N)}{2^{\ell_1}\varepsilon(p,\ell_1)} \approx \sqrt{N}\exp(-c(p)\log N).$$

□

Impact of the Variations of p. The optimal value of p for large N might not be the same as for $N = 1$. In practice when $\mathcal{T}(p, \ell_1)$ vary slowly with p (parameters corresponding to encryption) the behavior of Proposition 5 can be extended to the workfactor and, as long as N is not too large, we have

$$\mathrm{WF}_{\mathrm{ISD}}^{(N)}(n, r, w) \approx \frac{\mathrm{WF}_{\mathrm{ISD}}(n, r, w)}{N^{\gamma}} \text{ where } \gamma \approx \frac{1}{2} - 0.721 \frac{w - p}{r - \ell_1 - \frac{w-p-1}{2}} \tag{9}$$

where p and ℓ_1 are the optimal parameters of the algorithm when $N = 1$. For parameters corresponding to digital signature and hash function, the algorithm does not seem to take full benefit of multiple instances.

(n, r, w)	$\log_2 N$	p	ℓ	$\mathrm{WF}_{\mathrm{ISD}}^{(N)}$	observed γ	expected γ
$(4096, 540, 45)$	0	12	60	128.4	–	–
$(4096, 540, 45)$	40	12	80	110.5	0.4486	0.4487
$(4096, 540, 45)$	83.7	10	94	91.6	0.4398	0.4487
$(2048, 352, 32)$	0	6	30	81.0	–	–
$(2048, 352, 32)$	40	7	54	63.4	0.4403	0.4394
$(2048, 352, 32)$	51.4	7	60	58.8	0.4324	0.4394
$(2^{20}, 180, 11)$	0	10	94	87.8	–	–
$(2^{20}, 180, 11)$	40	6	79	79.6	0.2038	0.4856
$(2^{20}, 180, 11)$	70.3	4	76	74.6	0.1875	0.4856
$(2^{21}, 1024, 256)$	0	16	144	144.0	–	–
$(2^{21}, 1024, 256)$	40	6	79	141.5	0.0640	0.2724
$(2^{21}, 1024, 256)$	117.6	4	76	137.1	0.0597	0.2724

Table 5. Decoding N instances

5.3 Unlimited Number of Instances

We will denote

$$\mathrm{WF}_{\mathrm{ISD}}^{(\infty)}(n, r, w) = \min_{N,p,\ell} \mathcal{T}_N(p, \ell)$$

and we wish to compare this cost with $\mathrm{WF}_{\mathrm{ISD}}(n, r, w)$. If we assume that N unlimited, the best strategy for the attacker is to take exactly

$$N = \frac{1}{\varepsilon(p, \ell)\binom{k+\ell}{p}} = \frac{\min\left(2^r, \binom{n}{w}\right)}{\binom{r-\ell}{w-p}\binom{k+\ell}{p}}$$

instances, in which case (assuming $K_0 = 0$, see the discussion in §5.1) the complexity is

$$\mathcal{T}_\infty(p, \ell) = \frac{2\sqrt{K_1 K_2}}{\sqrt{\varepsilon(p, \ell)}} + \frac{K_3}{2^\ell \varepsilon(p, \ell)}$$

The minimal value is reached, up to a constant factor, when $\ell = \ell_\infty(p)$ such that

$$\ell_\infty(p) = \log_2\left(\frac{K_3}{2\sqrt{K_1 K_2 \varepsilon(p, \ell_\infty(p))}}\right).$$

Interestingly $\ell_\infty(p)$ is increasing with p and so is the complexity $\mathcal{T}(p, \ell_\infty(p))$. We thus want to choose p as small as possible. On the other hand, we have $|W_1||W_2| = \binom{k+\ell}{p}$ and $|W_2|$ must be a positive integer which limits the decrease of p. We must have

$$|W_1| \le \binom{k+\ell}{p} \Rightarrow \sqrt{\frac{K_2}{K_1 \varepsilon(p,\ell)}} \le \binom{k+\ell}{p},$$

with equality for the optimal p. Finally the optimal pair (p, ℓ) is the unique one such that we have simultaneously

$$\ell = \log_2\left(\frac{K_3}{2\sqrt{K_1 K_2}}\sqrt{\frac{\min\left(2^r, \binom{n}{w}\right)}{\binom{r-\ell}{w-p}}}\right) = \log_2\left(\frac{K_3}{2K_2}\binom{k+\ell}{p}\right).$$

An Estimate of the Improvement. Let p is the optimal value obtained above with an unlimited number of instances. In that case (we take $K_0 = 0$, $K_1 = K_2 = 1$, $K_3 = 2$)

$$\ell_1 = \log_2 \sqrt{\binom{k+\ell_1}{p}} \text{ and } \ell_\infty = \log_2 \binom{k+\ell_\infty}{p}.$$

Keeping the first order variations we have $\ell_\infty \approx 2\ell_1$. From Proposition 5 we have

$$\log_N \frac{\mathcal{T}(p,\ell_1)}{\mathcal{T}_\infty(p,\ell_\infty)} \approx \frac{1}{2} - c(p) \text{ where } c(p) = 0.721\frac{w-p}{r-\ell_1}$$

where $N \approx \mathcal{T}_\infty(p, \ell_\infty) \approx 2^{\ell_\infty}$. Thus

$$\mathcal{T}(p,\ell_1) \approx \mathcal{T}_\infty(p,\ell_\infty)^{\frac{3}{2}-c(p)}$$

Proposition 6 *For a given p, we have*

$$\frac{\log \mathcal{T}(p,\ell_1)}{\log \mathcal{T}_\infty(p,\ell_\infty)} \approx \frac{2}{3} + \frac{4}{9}c(p) \textit{ where } c(p) = \frac{1}{2\ln 2}\frac{w-p}{r-\ell_1-\frac{w-p-1}{2}}.$$

Coming back to the single instance case, and assuming that $\mathcal{T}(p, \ell_1)$ varies very slowly with p, we may assume that $\mathrm{WF}_{\mathrm{ISD}}(n, r, w) \approx \mathcal{T}(p, \ell_1)$. This means that when an attacker has access to an unlimited number of instances and needs to decode one of them only, the decoding exponent is multiplied by a quantity, slightly larger than 2/3, close to the one given in the above proposition.

$$\mathrm{WF}_{\mathrm{ISD}}^{(\infty)}(n,r,w) \approx \mathrm{WF}_{\mathrm{ISD}}(n,r,w)^\beta \text{ where } \beta \approx \frac{2}{3} + 0.321\frac{w-p}{r-\ell_1-\frac{w-p-1}{2}} \quad (10)$$

where p and ℓ_1 are the optimal parameters of the algorithm when $N = 1$.

We can observe that in Table 6, as for formula (9) and Table 5, the behavior is close to what we expect when encryption is concerned (when w is significantly smaller than the Gilbert-Varshamov distance). For parameter corresponding to the signature there is a gain but not as high as expected. For parameter corresponding to the hash function, multiple instances does not seem to provide a big advantage.

	$\log_2(\mathrm{WF}_{\mathrm{ISD}})$			$\log_2(\mathrm{WF}_{\mathrm{ISD}}^{(\infty)})$		obs.	exp.
(n, r, w)	p	ℓ		p	$= \ell$	β	β
$(2048, 352, 32)$	6	30	81.0	6.01	55.2	.682	.694
$(2048, 781, 71)$	6	29	100.7	8.20	69.2	.688	.696
$(4096, 252, 21)$	10	52	80.4	5.27	55.3	.688	.685
$(4096, 540, 45)$	12	60	128.4	9.00	88.0	.685	.689
$(8192, 416, 32)$	15	81	128.8	8.10	89.2	.693	.683
$(2^{16}, 144, 11)$	10	75	70.2	3.69	55.1	.785	.671
$(2^{16}, 160, 12)$	11	81	79.4	4.16	61.7	.777	.671
$(2^{18}, 162, 11)$	10	85	78.9	3.77	63.7	.808	.671
$(2^{20}, 180, 11)$	10	94	87.8	3.83	72.3	.824	.670
$(5 \cdot 2^{18}, 640, 160)$	10	91	91.8	4.45	84.8	.924	.768
$(7 \cdot 2^{18}, 896, 224)$	14	126	126.6	6.12	117.6	.929	.768
$(2^{21}, 1024, 256)$	16	144	144.0	6.96	134.0	.930	.768
$(23 \cdot 2^{16}, 1472, 368)$	24	206	205.9	10.48	191.7	.931	.768
$(31 \cdot 2^{16}, 1984, 496)$	32	275	275.4	14.01	257.2	.934	.767

Table 6. Workfactor with unlimited number of instances with the same code parameters as in Table 2

6 Conclusion

Decoding one out of many with collision decoding provides a significant advantage to an attacker. For the digital signature scheme, the threat is real because the attacker can create many syndromes by hashing many messages (favorable to him), however what we gain with ISD is less than what Bleichenbacher obtained with GBA. Anyway it is possible to completely avoid those attacks by signing several related syndromes (see [Fin10]).

For very large values of w (used for instance in FSB) we have seen that the attack is not so threatening, moreover the actual FSB [AFG$^+$08] or RFSB [BLPS11] use regular words and using ISD threatens an idealized version used for the security proofs. Decoding regular words is harder, and the question of how to decode one out of many and how to use it for an attack is still open.

Finally, for public-key encryption, when w is significantly smaller than the Gilbert-Varshamov distance, we take the full force of the attack. If there is a scenario where an attacker has access to many cryptograms and is satisfied by decoding only one of them, then there is a threat. We consider two scenarios

– the encryption scheme is used (often) to exchange session keys,
– the encryption scheme is used to encrypt a long stream of data.

Note that the attacker will only decrypt a single block, still we wish to avoid that. In the first scenario it is advisable to estimate the total number of session keys that will be used in a public-key lifetime and to increase the security parameters according to the result of the present study. The second scenario is plausible because code-based encryption is very fast, but in that case, it is enough to introduce some kind of chaining between encrypted blocks (which was advisable anyway) to counter the attack. Decrypting a single block will then be of no use to the attacker.

References

[AFG+08] D. Augot, M. Finiasz, Ph. Gaborit, S. Manuel, and N. Sendrier. SHA-3 proposal: FSB. Submission to the SHA-3 NIST competition, 2008.

[Bar98] A. Barg. Complexity issues in coding theory. In V. S. Pless and W. C. Huffman, editors, *Handbook of Coding theory*, volume I, chapter 7, pages 649–754. North-Holland, 1998.

[BLP08] D. 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.

[BLP11] D. Bernstein, T. Lange, and C. Peters. Smaller decoding exponents: Ball-collision decoding. In *Advances in Cryptology - CRYPTO 2011*, LNCS. Springer, 2011. To appear.

[BLPS11] Daniel J. Bernstein, Tanja Lange, Christiane Peters, and Peter Schwabe. Really fast syndrome-based hashing. Cryptology ePrint Archive, Report 2011/074, 2011. http://eprint.iacr.org/.

[BMvT78] E. R. Berlekamp, R. J. McEliece, and H. C. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3), May 1978.

[CC98] 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.

[CFS01] 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.

[CJ04] Jean-Sebastien Coron and Antoine Joux. Cryptanalysis of a provably secure cryptographic hash function. Cryptology ePrint Archive, Report 2004/013, 2004. http://eprint.iacr.org/.

[CP91] 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-Verlag, 1991.

[Dum91] I. Dumer. On minimum distance decoding of linear codes. In *Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory*, pages 50–52, Moscow, 1991.

[Fin10] M. Finiasz. Parallel-CFS: Strengthening the CFS McEliece-based signature scheme. In A. Biryukov, G. Gong, and D.R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *LNCS*, pages 159–170. Springer, 2010.

[FS96] 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.

[FS09] 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.

[GG07] P. Gaborit and M. Girault. Lightweight code-based identification and signature. In *IEEE Conference, ISIT'07*, pages 191–195, Nice, France, July 2007. IEEE.

[GLS07] P. Gaborit, C. Laudaroux, and N. Sendrier. Synd: a very fast code-based stream cipher with a security reduction. In *IEEE Conference, ISIT'07*, pages 186–190, Nice, France, July 2007. IEEE.

[JJ02] T. Johansson and F. Jönsson. On the complexity of some cryptographic problems based on the general decoding problem. *IEEE-IT*, 48(10):2669–2678, October 2002.

[LB88] P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In C. G. Günther, editor, *Advances in Cryptology - EUROCRYPT'88*, volume 330 of *LNCS*, pages 275–280. Springer-Verlag, 1988.

[Leo88] J. S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, September 1988.

[McE78] 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.

[Nie86] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Prob. Contr. Inform. Theory*, 15(2):157–166, 1986.

[OS09] R. Overbeck and N. Sendrier. Code-based cryptography. In D. Bernstein, J. Buchmann, and E. Dahmen, editors, *Post-Quantum Cryptography*, pages 95–145. Springer, 2009.

[Pet11] C. Peters. *Curves, Codes, and Cryptography.* PhD thesis, Technische Universiteit Eindhoven, 2011.

[Pra62] E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions*, IT-8:S5–S9, 1962.

[Saa07] 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.

[Ste89] 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-Verlag, 1989.

[Ste93] 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-Verlag, 1993.

[Vér97] P. Véron. Improved identification schemes based on error-correcting codes. *AAECC*, 8(1):57–69, January 1997.

[Wag02] D. Wagner. A generalized birthday problem. In M. Yung, editor, *CRYPTO'02*, volume 2442 of *LNCS*, pages 288–303. Springer, 2002.