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 techniques 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, for the parameter range corresponding to the McEliece encryption scheme, 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 the number of security bits is divided by a number slightly smaller than 3/2 (but larger than 1). Finally we give indications on how to counter those attacks.

1 Introduction

Code-based cryptography has attracted a lot of interest in the past few years, accompanying the rise of post-quantum cryptography. It allows public-key encryption scheme [21,22], zero-knowledge protocols [28,29,16], digital signature [11], hash functions [1,7], stream ciphers [15,17] 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. This problem is NP-hard [4], and in fact, the parameter selection for those systems is based on the best knows decoding techniques, usually the collision decoding [27] and its variants, and sometimes the generalized birthday algorithm (GBA) [8,30].

In this work, we consider the case where the attacker is given many instances of the decoding problem for the same linear code and wishes to solve only one of them. Bleichenbacher's attack against [11] (unpublished but described for instance in [23]) is a variant of GBA which offers a theoretical speedup of $\sqrt{N}$ when the attacker tries to sign one out of N messages. The cost of the attack will drop from T initially to $\max(T/\sqrt{N}, N)$ whose minimal value is $T^{2/3}$ (when $N = T^{2/3}$). A variant of ISD for multiple instances has been proposed [18], but its cost analysis does not allow an easy measure of the gain.

We consider in this paper a modification of ISD (similar to [18]) with a complete cost analysis. We will show that, when the number of errors to decode is smaller than the Gilbert-Varshamov distance[1] (corresponding to McEliece's or

[1] The (binary) Gilbert-Varshamov distance is the largest integer d_0 such that $\binom{n}{d_0} \leq 2^r$.

B.-Y. Yang (Ed.): PQCrypto 2011, LNCS 7071, pp. 51–67, 2011.

Niederreiter's encryption schemes), collision decoding can be adapted to save a factor $N^{0.5-c}$ (for some small positive c) 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 $2/3 + c'$ (for some small positive c'). In other words, the number of security bits (i.e. the log in base 2 of the cost of the best attack) is divided by some number close (but smaller to) 1.5.

We will first analyze an abstract variant of ISD, similar to the one of [14]. We will then show how this algorithm and its analysis can be extended to the case of many instances and provide some estimates of what this modified algorithm can gain. This new attack constitutes a threat which must be considered. We briefly explain in the conclusion how to completely avoid it. The countermeasures are 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 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 [21,22] and digital signature [11], zero-knowledge protocols based on codes [28,29,16], hash-function [1], PRNG and stream ciphers [15,17] and many others, decoding is the most threatening attack and therefore is a key point in the parameter selection.

2.1 Generic Decoding Algorithms

The most ancient technique for addressing CSD in cryptology is Information Set Decoding (ISD). It can be traced back to Prange [25]. The variants useful today in cryptology all derive more or less from Stern's algorithm [27], which we will call collision decoding, following [6,24]. It was implemented (with various improvements) in [9] then in [5] which reports the first successful attack on the original parameter set. General lower bounds were proposed [14]. The last published variant is ball-collision decoding [6] which features a better decoding exponent than collision decoding.

The other main technique is the Generalized Birthday Algorithm (GBA) [30] (order 2 GBA was previously published in [8]). The first use of GBA for decoding was proposed in [10] for attacking an early version of FSB [2]. It is sometimes faster than ISD.

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 [26].

For multiple instances the situation is not known precisely, but in one case at least (namely Bleichenbacher's attack against CFS signature scheme) GBA with multiple instances has become the most efficient attack. This was a motivation to consider whether a similar improvement was possible with ISD.

2.2 Decoding One Out of Many Instances

In this work we will consider the scenario where the attacker has many instances (H, s, w) at disposal 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 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 [23]) for attacking the digital signature CFS. In practice, the attacker builds a large number N of instances of a decoding problem (corresponding to N favorable messages) solves one of them with an order 2 GBA with a speedup of $\sqrt{N}$ compared with the decoding of a single instance with a birthday attack. This reduces the order of magnitude of the cost for forging a signature from $O(2^{r/2})$ to $O(2^{r/3})$. A variant of CFS resistant to this attack was recently published [13].

An attempt at using ISD with multiple instances was already made in [18]. 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 [19,20], J. Stern describes in [27] an algorithm to solve CSD. We present in Algorithm 1 a generalized version, similar to the one presented in [14], which acts on the parity check matrix H_0 of the code (instead of the generator matrix). The partial Gaussian elimination of H_0P consists in finding U and H (and H', H'') such that[2]

Algorithm 1. Generalized ISD algorithm

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) {         $P \leftarrow$ random $n \times n$ permutation matrix
        {         $(H', H'', U) \leftarrow$ PartialGaussElim$(H_0P)$          // as in (1)
        {         $s \leftarrow s_0U^T$
                  $e \leftarrow$ isd_loop$(H', H'', s)$
              while $e =$ 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) {         $i \leftarrow e_1H'^T$, $s''_1 \leftarrow e_1H''^T$
            {         write$(e_1, s''_1, i)$          // stores $(e_1, s''_1)$ at index $i$
                  for all $e_2 \in W_2$
    (ISD 2) {         $i \leftarrow s' + e_2H'^T$, $s''_2 \leftarrow s'' + e_2H''^T$
            {         Elts $\leftarrow$ read$(i)$     // extracts the elements stored at index $i$
                      for all $(e_1, s''_1) \in$ Elts
        (ISD 3) {         if wt$(s''_1 + s''_2) = w - p$
                {             return $e_1 + e_2$                      (SUCCESS)
                  return FAIL                                         (FAIL)
```

$$UH_0P = H = \begin{array}{c|c|c|} & r-\ell & k+\ell \\ \cline{2-3} & \begin{matrix}1 & & \\ & \ddots & \\ & & 1\end{matrix} & H'' \\ \cline{2-3} \ell & 0 & H' \\ \cline{2-3} \end{array}, \quad s^T = Us_0^T = \begin{array}{|c|} \hline s''^T \\ \\ \hline s'^T \\ \hline \end{array} \qquad (1)$$

[2] If the first $r - \ell$ columns are dependent, we change P.

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

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 quantity allowing us to compare algorithms and to measure the impact of decoding one out of many. Any reasonably fixed polynomial time (in n) "elementary operation" will serve that purpose.

3.1 A Preview of the Analysis

When there is a single solution to CSD ("small" w, corresponding to encryption) we can provide some intuition on the significance of the parameters.

Significance of W_1 and W_2. Given p and ℓ, we would like $W_1 + W_2 = \{e_1 + e_2 \mid (e_1, e_2) \in W_1 \times W_2\}$ to contain as many distinct elements of $\mathcal{S}_{k+\ell}(\mathbf{0}, p)$ as possible, but no (or not too many) duplicate sums[3]. Typically the elements of W_1 and those of W_2 are chosen with distinct supports, for instance in [12]

$$W_1 = \left\{(e, 0) \mid e \in \mathcal{S}_{\frac{k+\ell}{2}}(\mathbf{0}, \tfrac{p}{2})\right\} \text{ and } W_2 = \left\{(0, e) \mid e \in \mathcal{S}_{\frac{k+\ell}{2}}(\mathbf{0}, \tfrac{p}{2})\right\}$$

(assuming p and $k + \ell$ are even). A proper choice of W_1 and W_2 will allow us to find most solutions $e' \in \mathrm{CSD}(H', s', p)$ (see (1) for the notations) for a relatively moderate cost (exploring $W_1 \times W_2$ uses the birthday paradox and essentially consists in exploring W_1 then W_2).

Significance of p and ℓ. The optimal size of W_1 and W_2 depends on p and ℓ. Given p, the best value for ℓ keeps a balance between the costs of the various steps of the algorithm and it is best to choose $2^\ell \approx |W_1| = |W_2|$. There is no easy interpretation of the optimal p, but an easy analysis shows that the extremal cases $p = 0$ or w (either H' or H'' vanishes in (1)) are not optimal. So there has to be an optimal value for p between 0 and w.

3.2 Links With the Other Variants of Collision Decoding

Information set decoding is an old decoding technique [25], the variants of interest today for cryptanalysis derive from Stern's collision decoding [27]. The algorithm we present here is closer to the "Punctured Split Syndrome Decoding" of Dumer [12,3]. Depending on how the sets W_1 and W_2 are chosen, we

[3] That is $(e_1, e_2) \neq (e'_1, e'_2)$ in $W_1 \times W_2$ such that $e_1 + e_2 = e'_1 + e'_2$.

may obtain any known variant, including the recent ball-collision decoding [6]. 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 (10) and (11) 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).

We are given all the algorithm parameters n, r, w, p, ℓ, W_1, and W_2. For computing probabilities (and thus cost estimates) we will make the usual random coding assumption (pseudo-randomness of syndromes) and also assume that Algorithm 1 runs on instances which have a solution (this makes sense for a cryptanalysis). We also 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.

All the formulas we will give in the rest of the paper will depend of one fundamental quantity denoted $\varepsilon(p,\ell)$. It is equal to 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. The following estimates helps to understand how it varies with p and ℓ

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

Proof. (of equation (2), sketch) We consider one particular execution of isd_loop and use all the notations of the algorithm. Given H' and H'', for any $e' \in \mathcal{S}_{k+\ell}(\mathbf{0}, p)$ we count how many $s = (s', s'')$ are such that e' is a valid output of isd_loop(H', H'', s). We must have $s' = e'H'^T$ and $s'' \in \mathcal{S}_{r-\ell}(e'H''^T, w-p)$, that is $\binom{r-\ell}{w-p}$ "good" values of s (1 for s' multiplied by $\binom{r-\ell}{w-p}$ for s''). Because Algorithm 1 is executed on an instance having solutions, we must have $s \in \mathcal{U} = \{eH^T \mid e \in \mathcal{S}_n(\mathbf{0}, w)\}$. It follows that $\varepsilon(p, \ell) = \binom{r-\ell}{w-p}/|\mathcal{U}|$. Within our assumptions, the set $\mathcal{U}$ can viewed as a set of $\binom{n}{w}$ randomly chosen elements of $\{0, 1\}^r$ and thus *on average*

$$|\mathcal{U}| = 2^r \left(1 - \left(1 - \frac{1}{2^r}\right)^{\binom{n}{w}}\right) \approx \min\left(2^r, \binom{n}{w}\right)$$

from which we deduce the expression (2) of $\varepsilon(p, \ell)$. □

Let $W_1 + W_2 = \{e_1 + e_2 \mid (e_1, e_2) \in W_1 \times W_2\}$, we also introduce

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

the probability of one particular execution of isd_loop succeed. Note that within our assumptions $|W_1 + W_2| = |W_1 \times W_2| = |W_1||W_2|$.

Proposition 1. *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*

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

elementary operations on 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 on average $|W_1||W_2|/2^\ell$ times. Thus if π_3 is its probability of success, we have (remember $|W_1 + W_2| = |W_1||W_2|$)

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

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

A consequence of this proposition is that the minimal cost for Algorithm 1 is obtained when $|W_2|$ is maximal (everything else being fixed), that is 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) are equal, 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 [14]. We have

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

where

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

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 cost for the Gaussian elimination step), the cost estimate becomes

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

and because the first term is increasing and the second is decreasing with ℓ (for parameters of cryptologic interest), for all p we have $\mathcal{T}(p,\ell_1)/2 \leq \min_\ell \mathcal{T}(p,\ell) \leq \mathcal{T}(p,\ell_1)$ 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

$$\ell_1 = \log_2\left(\frac{K_3}{2K_2}\mathcal{L}(p,\ell_1)\right) = \log_2\left(\frac{K_3}{2\sqrt{K_1K_2}}\sqrt{\frac{\mathcal{P}(p,\ell_1)}{\varepsilon(p,\ell_1)}}\right). \tag{8}$$

The lower bound is $\mathrm{WF}_{\mathrm{ISD}}(n,r,w) \geq \min_p \mathcal{T}(p,\ell_1)/2$ 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 very tight if the Gaussian elimination cost is negligible (which is often the case in practice, see Table 2). Numbers in Table 2 may seem different from other estimates [5,14]. This difference comes from the fact that we consider column operations rather than binary operations. In fact they are very close.

4.2 Some Numbers

For Table 3 we 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. The cost for (ISD 1) and (ISD 2) can be reduced to 1 by "reusing additions", as explained in [5]. 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 on average $2(w-p)$ additional bits [5] 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.

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.

(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

4.3 Variations with the Parameter p

With (8), we have 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 $\mathrm{CSD}(H_0, \mathcal{S}_0, w)$ for a set of $\mathcal{S}_0$ of N independent syndromes which all have a solution. We describe a procedure for that in Algorithm 4. This algorithm is very similar to Algorithm 1. The differences are related to the set of syndromes $\mathcal{S}_0$. In the block (DOOM 0) we compute $\mathcal{S} = \{s_0 U^T \mid s_0 \in \mathcal{S}_0\}$ instead of just $s = s_0 U^T$ and in the procedure doom_loop, the second loop we run through $W_2 \times \mathcal{S}$ instead of W_2. It is still optimal to have $W_1 + W_2$ close to $\mathcal{S}_{k+\ell}(\mathbf{0}, p)$, but instead of $|W_1| = |W_2|$ in Algorithm 1, it is better now to choose $|W_1| = |W_2 \times \mathcal{S}| = N|W_2|$.

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

$(n, r, w) = (4096, 540, 45)$												
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

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

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

the probability for one execution of doom_loop to succeed. We have a statement very similar to Proposition 1.

Proposition 2. *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*

$$\approx \mathcal{T}_N(p, \ell) = \frac{K_0}{\mathcal{P}_N(p, \ell)} + \frac{K_1|W_1|}{\mathcal{P}_N(p, \ell)} + \frac{K_2}{|W_1|\varepsilon(p, \ell)} + \frac{K_3}{2^\ell \varepsilon(p, \ell)} \tag{9}$$

elementary operations on average.

We omit the proof which is similar to the proof of Proposition 1 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 $s_o U^T$ 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 $N \leq 1/\varepsilon(p, \ell)\binom{k+\ell}{p}$ (with larger N just reading the instances would be the bottleneck, so we discard that possibility) 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 $s_o U^T$ will require

Algorithm 4. DOOM ISD algorithm

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)  {       $P \leftarrow$ random $n \times n$ permutation matrix
          {       $(H', H'', U) \leftarrow \mathrm{PartialGaussElim}(H_0 P)$          // as in (1)
          {       $\mathcal{S} \leftarrow \{s_0 U^T \mid s_0 \in \mathcal{S}_0\}$
                  $e \leftarrow$ 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)  {           $i \leftarrow e_1 H'^T$, $s_1'' \leftarrow e_1 H''^T$
          {           write$(e_1, s_1'', i)$                  // stores $(e_1, s_1'')$ at index $i$
                  for all $e_2 \in W_2$
                      for all $s = (s', s'') \in \mathcal{S}$
(DOOM 2)  {               $i \leftarrow s' + e_2 H'^T$, $s_2'' \leftarrow s'' + e_2 H''^T$
          {               Elts $\leftarrow$ read$(i)$     // extracts the elements stored at index $i$
                          for all $(e_1, s_1'') \in$ Elts
(DOOM 3)  {                   if wt $(s_1'' + s_2'') = w - p$
          {                       return $e_1 + e_2$                          (SUCCESS)
                  return FAIL                                                  (FAIL)
```

about $Nr/\log_2 N$ column operations. For the extremal values of N of §5.3 (the case most favorable to the attacker), assuming $K_1 = K_2 = K_3/2 = 1$, we have $\mathcal{P}_n(p, \ell) = 1$ and a complexity $\approx Nr/\log_2 N + 2^{\ell+2}$ with $N = 2^{2\ell}/\binom{k+\ell}{p} \leq 2^\ell$. 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 minimizes the formula for the cost (this value, by the way, is not necessarily an integer and does not correspond to a practical implementation) we have a complexity of the form $2^\ell(r/\ell + 4)$ and we cannot 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

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

and the gain we wish to estimate is the ratio

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

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 $K_0 = 0$)

$$\mathcal{T}_N(p,\ell) = \frac{2\sqrt{K_1 K_2}}{\sqrt{N\binom{k+\ell}{p}}\varepsilon(p,\ell)} + \frac{K_3}{2^\ell \varepsilon(p,\ell)}.$$

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 3. *For a given p, we have*

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

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

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

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

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

□

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 3 can be extended to the workfactor and, as long as N is not too large, we have

$$\mathrm{WF}_{\mathrm{ISD}}^{(N)}(n, r, w) = \frac{\mathrm{WF}_{\mathrm{ISD}}(n, r, w)}{N^\gamma} \text{ where } \gamma \approx \frac{1}{2} - 0.721 \frac{w - p}{r - \ell_1 - \frac{w-p-1}{2}} \quad (10)$$

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.

Table 5. Decoding N 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

5.3 Unlimited Number of Instances

We assume that the attacker can let N grow indefinitely. Because any algorithm must at least read its input there is a limit to the growth of N. By "unlimited" we mean that the attacker has reached this limit (whatever it is). 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 $\mathrm{WF}_{\mathrm{ISD}}(n, r, w)$. The best strategy for the attacker is to take a number of instances equal to

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

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

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

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 3 we have

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

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 4. *For a given p, we have*

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

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.

$$\mathrm{WF}_{\mathrm{ISD}}^{(\infty)}(n,r,w) = \mathrm{WF}_{\mathrm{ISD}}(n,r,w)^\beta \text{ where } \beta \approx \frac{2}{3} + 0.321 \frac{w-p}{r - \ell_1 - \frac{w-p-1}{2}} \quad (11)$$

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 (10) 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 parameters for code-based signature schemes there is a gain but not as high as expected. For parameters for code-based hashing, multiple instances does not seem to provide a big advantage. The values of p and ℓ given in the fifth and sixth columns are real numbers which minimize the formula for $\log_2(\mathrm{WF}_{\mathrm{ISD}}^{(\infty)})$. In an implementation they must be integers and the real cost will be (marginally) different.

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

	$\log_2(\mathrm{WF}_{\mathrm{ISD}})$			$\log_2(\mathrm{WF}_{\mathrm{ISD}}^{(\infty)})$		observed	expected
(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

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 syndromes (see [13]).

For very large values of w (used for instance in hashing) we have seen that the attack is not so worrying, moreover the actual FSB [1] or RFSB [7] 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, when w is significantly smaller than the Gilbert-Varshamov distance (for public-key encryption) there is a gain. If the attacker has access to many cryptograms and is satisfied by decoding only one of them, the present work must be taken into account. We consider two scenarios: (1) the encryption scheme is used to exchange session keys, and (2) the encryption scheme is used to encrypt a long stream of data. In the first scenario the number of session keys in a public key lifetime must be used to select 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 to counter the attack. Decrypting a single block will then be of no use to the attacker.

Acknowledgements. The author would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

1. Augot, D., Finiasz, M., Gaborit, P., Manuel, S., Sendrier, N.: SHA-3 proposal: FSB. Submission to the SHA-3 NIST Competition (2008), `http://www-rocq.inria.fr/secret/CBCrypto/index.php?pg=fsb`
2. Augot, D., Finiasz, M., Sendrier, N.: A fast provably secure cryptographic hash function. Cryptology ePrint Archive, Report 2003/230 (2003), `http://eprint.iacr.org/`
3. Barg, A.: Complexity issues in coding theory. In: Pless, V., Huffman, W. (eds.) Handbook of Coding Theory, vol. I, ch. 7, pp. 649–754. North-Holland (1998)
4. Berlekamp, E., McEliece, R., van Tilborg, H.: On the inherent intractability of certain coding problems. IEEE Trans. on Information Theory 24(3) (May 1978)
5. Bernstein, D.J., Lange, T., Peters, C.: Attacking and Defending the Mceliece Cryptosystem. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 31–46. Springer, Heidelberg (2008)
6. Bernstein, D.J., Lange, T., Peters, C.: Smaller Decoding Exponents: Ball-Collision Decoding. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 743–760. Springer, Heidelberg (2011)
7. Bernstein, D.J., Lange, T., Peters, C., Schwabe, P.: Really Fast Syndrome-Based Hashing. In: Nitaj, A., Pointcheval, D. (eds.) AFRICACRYPT 2011. LNCS, vol. 6737, pp. 134–152. Springer, Heidelberg (2011)
8. Camion, P., Patarin, J.: The Knapsack Hash Function Proposed at Crypto'89 can be Broken. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 39–53. Springer, Heidelberg (1991)
9. 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. IEEE Trans. on Information Theory 44(1), 367–378 (1998)
10. Coron, J.S., Joux, A.: Cryptanalysis of a provably secure cryptographic hash function. Cryptology ePrint Archive, Report 2004/013 (2004), `http://eprint.iacr.org/`
11. Courtois, N.T., Finiasz, M., Sendrier, N.: How to Achieve a McEliece-Based Digital Signature Scheme. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 157–174. Springer, Heidelberg (2001)
12. Dumer, I.: On minimum distance decoding of linear codes. In: Proc. 5th Joint Soviet-Swedish Int. Workshop Inform. Theory, Moscow, pp. 50–52 (1991)
13. Finiasz, M.: Parallel-CFS: Strengthening the CFS McEliece-Based Signature Scheme. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 159–170. Springer, Heidelberg (2011)

14. Finiasz, M., Sendrier, N.: Security Bounds for the Design of Code-Based Cryptosystems. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 88–105. Springer, Heidelberg (2009)
15. Fischer, J.B., Stern, J.: An Efficient Pseudo-Random Generator Provably as Secure as Syndrome Decoding. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 245–255. Springer, Heidelberg (1996)
16. Gaborit, P., Girault, M.: Lightweight code-based identification and signature. In: IEEE Conference, ISIT 2007, pp. 191–195. IEEE, Nice (2007)
17. Gaborit, P., Laudaroux, C., Sendrier, N.: Synd: a very fast code-based stream cipher with a security reduction. In: IEEE Conference, ISIT 2007, pp. 186–190. IEEE, Nice (2007)
18. Johansson, T., Jönsson, F.: On the complexity of some cryptographic problems based on the general decoding problem. IEEE-IT 48(10), 2669–2678 (2002)
19. Lee, P.J., Brickell, E.F.: An Observation on the Security of McEliece's Public-Key Cryptosystem. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 275–280. Springer, Heidelberg (1988)
20. Leon, J.: A probabilistic algorithm for computing minimum weights of large error-correcting codes. IEEE Trans. on Information Theory 34(5), 1354–1359 (1988)
21. McEliece, R.: A public-key cryptosystem based on algebraic coding theory. DSN Prog. Rep., Jet Prop. Lab., California Inst. Technol., Pasadena, CA pp. 114–116 (January 1978)
22. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Prob. Contr. Inform. Theory 15(2), 157–166 (1986)
23. Overbeck, R., Sendrier, N.: Code-based cryptography. In: Bernstein, D., Buchmann, J., Dahmen, E. (eds.) Post-Quantum Cryptography, pp. 95–145. Springer, Heidelberg (2009)
24. Peters, C.: Curves, Codes, and Cryptography. Ph.D. thesis, Technische Universiteit Eindhoven (2011)
25. Prange, E.: The use of information sets in decoding cyclic codes. IRE Transactions IT-8, S5–S9 (1962)
26. Saarinen, M.J.: Linearization Attacks against Syndrome Based Hashes. In: Srinathan, K., Rangan, C.P., Yung, M. (eds.) INDOCRYPT 2007. LNCS, vol. 4859, pp. 1–9. Springer, Heidelberg (2007)
27. Stern, J.: A Method for Finding Codewords of Small Weight. In: Wolfmann, J., Cohen, G. (eds.) Coding Theory 1988. LNCS, vol. 388, pp. 106–113. Springer, Heidelberg (1989)
28. Stern, J.: A New Identification Scheme Based on Syndrome Decoding. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 13–21. Springer, Heidelberg (1994)
29. Véron, P.: Improved identification schemes based on error-correcting codes. AAECC 8(1), 57–69 (1997)
30. Wagner, D.: A Generalized Birthday Problem. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 288–303. Springer, Heidelberg (2002)