

# Generic Decoding of Restricted Errors

Sebastian Bitzer<sup>1</sup>, Alessio Pavoni<sup>2</sup>, Violetta Weger<sup>1</sup>, Paolo Santini<sup>2</sup>, Marco Baldi<sup>2</sup>, Antonia Wachter-Zeh<sup>1</sup>

<sup>1</sup>Technical University of Munich, Germany

{sebastian.bitzer, violetta.weger, antonia.wachter-zeh}@tum.de

<sup>2</sup>Marche Polytechnic University, Italy

s1101018@studenti.univpm.it, {p.santini, m.baldi}@staff.univpm.it

**Abstract**—Several recently proposed code-based cryptosystems base their security on a slightly generalized version of the classical (syndrome) decoding problem. Namely, in the so-called restricted (syndrome) decoding problem, the error values stem from a restricted set. In this paper, we propose new generic decoders, that are inspired by subset sum solvers and tailored to the new setting. The introduced algorithms take the restricted structure of the error set into account in order to utilize the representation technique efficiently. This leads to a considerable decrease in the security levels of recently published code-based cryptosystems.

## I. INTRODUCTION

With the recent advances in quantum technology, the search for quantum-secure cryptographic systems has become one of the most pressing challenges. In the NIST selection process of post-quantum cryptosystems launched in 2016, which has now reached the 4th round, some of the most promising candidates are based on algebraic coding theory, more precisely on the hardness of decoding a random linear code. A new trend in code-based cryptography is to base the security on the hardness of a slightly different problem, e.g. [7]. One such new decoding problem is a generalized decoding problem, where one restricts the error set. Some recent systems base their security on this decoding problem [1], [18].

In this paper, we provide new solvers for such settings, which are inspired by subset sum solvers [3]. The idea is to use the additive structure that can be found in the error set and to add only few elements to the search set, such that one achieves more representations of the elements in the error set and does not increase the search sizes too much. The connection between subset sum solvers and generic decoders has often been exploited (see, e.g., [4], [11]). We show the impact of these new attacks on systems to which they apply and how significantly their security levels decrease. As the restricted decoding problem is still very promising for cryptographic applications, the presented attacks should be considered in future proposals. For this purpose, we also shortly explain how the specific solvers can be generalized to any setting.

The paper is structured as follows: in Section II, we recall some basic notions of algebraic coding theory and introduce the required notation. In Section III, we introduce the restricted decoding problem and discuss some of its properties. We then present the new attacks in Section IV, comparing the approaches in different cases and computing new security levels for cryptosystems that are using the restricted decoding problem. Finally, Section V concludes the paper.

## II. PRELIMINARIES

Throughout this paper we denote by  $q$  a prime power and by  $\mathbb{F}_q$  a finite field of order  $q$ . We denote the identity matrix of size  $n$  by  $\text{Id}_n$ . For a set  $J$ , we denote by  $J_0 = J \cup \{0\}$ . For  $x \in [0, 1]$ , we denote by  $h(x)$  the binary entropy function. For  $n \geq k_1 + k_2$  we denote by  $\binom{n}{k_1, k_2} = \binom{n}{k_1 + k_2} \cdot \binom{k_1 + k_2}{k_1}$  the trinomial coefficient. Recall that

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log_2 \left( \binom{f_1(n)}{f_2(n)} \right) = F_1 h\left(\frac{F_2}{F_1}\right),$$

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log_2 \left( \binom{f_1(n)}{f_2(n), f_3(n)} \right) = F_1 g\left(\frac{F_2}{F_1}, \frac{F_3}{F_1}\right),$$

with  $F_i = \lim_{n \rightarrow \infty} \frac{f_i(n)}{n}$ ,  $h(x)$  the binary entropy function and  $g(x, y) = -x \log_2(x) - y \log_2(y) - (1 - x - y) \log_2(1 - x - y)$ .

A linear code  $\mathcal{C}$  is a  $k$ -dimensional subspace of  $\mathbb{F}_q^n$ . A linear code can be compactly represented either through a generator matrix  $\mathbf{G} \in \mathbb{F}_q^{k \times n}$  or through a parity-check matrix  $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ , which have the code as image or as kernel, respectively. We say that a linear code has rate  $R = k/n$ . We define  $\mathcal{C}_J$  as  $\mathcal{C}_J = \{c_J \mid c \in \mathcal{C}\}$ , where  $c_J$  is the projection of  $c$  on the coordinates indexed by  $J$ . For any  $x \in \mathbb{F}_q^n$ , we call  $s = x\mathbf{H}^\top \in \mathbb{F}_q^{n-k}$  a syndrome. A set  $I \subseteq \{1, \dots, n\}$  of size  $k$  is called an information set for  $\mathcal{C}$ , if  $|I| = |\mathcal{C}_I|$ . We say that a generator matrix, respectively a parity-check matrix, is in systematic form (with respect to the information set  $I$ ), if the columns of  $\mathbf{G}$  indexed by  $I$  form  $\text{Id}_k$ , respectively, if the columns of  $\mathbf{H}$  not indexed by  $I$  form  $\text{Id}_{n-k}$ . We endow the vector space  $\mathbb{F}_q^n$  with the Hamming metric: the Hamming weight of a vector  $x \in \mathbb{F}_q^n$  is given by the number of its non-zero entries, i.e.,

$$\text{wt}_H(x) = |\{i \in \{1, \dots, n\} \mid x_i \neq 0\}|,$$

which then induces a distance, as  $d_H(x, y) = \text{wt}_H(x - y)$ , for  $x, y \in \mathbb{F}_q^n$ .

## III. RESTRICTED SYNDROME DECODING PROBLEM

Throughout this paper, we consider the computational version of the following decisional problem.

**Problem 1** (Restricted Syndrome Decoding Problem (RSDP)). *Let  $g \in \mathbb{F}_q$  be an element of order  $z$  and define the error set  $\mathbb{E} = \{g^i \mid i \in \{1, \dots, z\}\}$ . Let  $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ ,  $w \in \mathbb{N}$  and  $s \in \mathbb{F}_q^{n-k}$ . Does there exist a vector  $e \in \mathbb{E}_0^n = (\mathbb{E} \cup \{0\})^n$  with  $\text{wt}_H(e) = w$ , such that  $s = e\mathbf{H}^\top$ ?*

If we choose  $g$  to be a primitive element of  $\mathbb{F}_q$  we can indeed recover the original syndrome decoding problem (SDP) in the Hamming metric. Thus, the problem directly inherits the NP-completeness. Even more is true, for a fixed  $g$ , the RSDP is still NP-complete, this follows directly from [20, Proposition 2].

**Corollary 1.** *The RSDP with a fixed  $\mathbb{E}$  is NP-complete.*

In addition to generalizing the classical SDP, the RSDP also covers the case  $\mathbb{E} = \{\pm 1\}$  considered in [1], by setting  $g = -1$ , respectively  $z = 2$ .

Also [18] considers a particular case of the RSDP. In their paper, they consider  $\mathbb{F}_q$ , where  $q = 6m + 1$  is a prime, that is the field allows for an element of order 6. They then consider the errors to live in  $\mathbb{E} = \{\pm 1, \pm g, \pm(g-1)\}$ . This corresponds to the RSDP with  $z = 6$ . In fact, since  $g$  is a root of  $x^6 - 1 = (x^3 - 1)(x^3 + 1)$  and  $g$  is not a root of  $(x^3 - 1)$  or  $(x^3 + 1)$ , we must have  $g^2 = g - 1$ .

From the uniqueness condition  $\binom{n}{w} z^w q^{k-n} \leq 1$ , we can easily observe that the restriction on the entries of the error vector  $e$  allows us to increase the Hamming weight of  $e$ , while still having a single solution to the RSDP with high probability. This is the main motivation for introducing the RSDP in cryptographic applications.

#### IV. SOLVERS FOR THE RSDP

There exist several algorithms that could potentially be applied to the RSDP, for example statistical decoders, such as [6], or attacks from lattice-based cryptography, such as [12]. In this paper, we focus on Information Set Decoding (ISD) algorithms combined with ideas from subset sum solvers, such as [3].

The history of ISD dates back to the algorithm of Prange [15] in 1962 and has resulted in several improvements (for an overview in the binary case see [14], [19]). Variants of Prange's ISD with smaller computational complexity are Stern/Dumer [8], [17], MMT [13] and BJMM [4], [10]. We only shortly recall their ideas in the following, before we adapt them to our setting. Given  $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ ,  $w \in \mathbb{N}$  and  $\mathbf{s} \in \mathbb{F}_q^{n-k}$ , one starts by choosing a set  $J \subseteq \{1, \dots, n\}$  of size  $k + \ell$ , for some positive integer  $\ell \leq n - k$ , which contains an information set. One then brings the parity-check matrix into systematic form, by performing Partial Gaussian Elimination (PGE), denoted by  $\mathbf{H}'$ , and performs the same operations on the syndrome. For simplicity, assume that the set  $J$  is chosen in the first  $k + \ell$  positions. Thus, we get the syndrome equations

$$\mathbf{e} \mathbf{H}'^\top = (\mathbf{e}_1 \quad \mathbf{e}_2) \begin{pmatrix} \text{Id}_{n-k-\ell} & \mathbf{0} \\ \mathbf{A}_1 & \mathbf{A}_2 \end{pmatrix} = (\mathbf{s}_1 \quad \mathbf{s}_2),$$

where  $\mathbf{A}_1 \in \mathbb{F}_q^{(k+\ell) \times (n-k-\ell)}$ ,  $\mathbf{A}_2 \in \mathbb{F}_q^{(k+\ell) \times \ell}$ ,  $\mathbf{e}_1 \in \mathbb{F}_0^{n-k-\ell}$ ,  $\mathbf{e}_2 \in \mathbb{F}_0^{k+\ell}$ ,  $\mathbf{s}_1 \in \mathbb{F}_q^{n-k-\ell}$  and  $\mathbf{s}_2 \in \mathbb{F}_q^\ell$ . One first solves for  $\mathbf{e}_2$ , assuming a weight  $v$ , i.e.,  $\mathbf{e}_2 \mathbf{A}_2^\top = \mathbf{s}_2$  and then checks if  $\mathbf{e}_1 = \mathbf{s}_1 - \mathbf{e}_2 \mathbf{A}_1^\top$  has the remaining weight  $w - v$  and entries in  $\mathbb{E}_0$ . To solve the smaller instance given by  $(\mathbf{A}_2, v, \mathbf{s}_2)$  one can use different approaches.

Before we start describing these approaches, let us fix some notation. To compare algorithms for fixed  $Q = \log_2(q)$ ,  $Z = \log_2(z)$ , we are interested in the asymptotic cost, that is, we write the cost as  $2^{F(R,W)n}$ , for some function  $F(R, W)$ , where  $R = \lim_{n \rightarrow \infty} \frac{k(n)}{n}$ ,  $W = \lim_{n \rightarrow \infty} \frac{w(n)}{n}$ . Since we have seen that it is enough to solve the smaller instance of weight  $v$  and length  $k + \ell$ , we also write  $L = \lim_{n \rightarrow \infty} \frac{\ell(n)}{n} \leq 1 - R$ ,  $V = \lim_{n \rightarrow \infty} \frac{v(n)}{n} \leq \min\{W, R + L\}$ , which are internal parameters and can hence be optimized. Then, the complexity of a decoder using the PGE setup is given by the following theorem, see, e.g., [14].

**Theorem 2.** *A generic decoder using PGE has time complexity  $2^{F(R,W)n}$ , with  $F(R, W) = N(R, W, L, V) + C(R, L, V)$ , where  $N(R, W, L, V)$  denotes the number of iterations, i.e.,*

$$h(W) - (R + L)h\left(\frac{V}{R+L}\right) - (1 - R - L)h\left(\frac{W-V}{1-R-L}\right)$$

and  $C(R, L, V)$  denotes the time complexity of solving the smaller instance, i.e., the time to enumerate all solutions of the smaller instance under the assumed weight distribution.

We compare our algorithms to different approaches, namely Stern/Dumer [8], [17] and BJMM [4], [10], which encompasses MMT as well [13], adapted to the new setting. We proceed by explaining how this adaption works.

A classical approach to enumerating all solutions of the small instance is performing a collision search. This technique was applied to hard knapsacks by Schroepel and Shamir [16] and adopted for the syndrome decoding problem by Stern and Dumer [8], [17]. In this approach one uses a set partition of  $\mathbb{E}_2$  into  $\mathbb{E}_2 = (\mathbf{x}_1, \mathbf{x}_2)$ , where both  $\mathbf{x}_i \in \mathbb{E}_0^{(k+\ell)/2}$  have weight  $v/2$ . One constructs lists containing such  $\mathbf{x}_i$  and by a collision search finds candidates  $\mathbf{e}_2$ . This merging process is called concatenation merge. For more details on the classical algorithm we refer to [8], [17] and for the adaption to [2].

**Lemma 3.** *The enumeration cost for the smaller instance of the restricted Stern/Dumer algorithm is given by  $C(R, L, V) = \max\{\frac{\Sigma}{2}, \Sigma - Q \cdot L\}$ , where*

$$\Sigma = (R + L)h\left(\frac{V}{R+L}\right) + ZV$$

is the asymptotic size of the search space, i.e., the set of all vectors that are well-formed, i.e., they satisfy the constraint under which the solutions of the small instance are enumerated.

As this is a well-known algorithm with the only change that the lists are taken in  $\mathbb{E}_0$ , rather than in  $\mathbb{F}_q$ , we omit the proof.

More in general, one can perform a concatenation merge of two lists  $\mathcal{L}_1, \mathcal{L}_2$ , of asymptotic size  $\Lambda = \lim_{n \rightarrow \infty} \frac{\log_2(|\mathcal{L}_i|)}{n}$ , requiring only that they satisfy a syndrome equation on  $u$  positions, with  $U = Q \cdot \lim_{n \rightarrow \infty} \frac{u(n)}{n}$ . This costs asymptotically

$$2^{n \max\{\Lambda, 2\Lambda - U\}}.$$

An alternative to this collision search is using the representation technique, which has proven efficient in solving the hard knapsacks [3], [11] and the classical SDP [4], [13].

Instead of a set partition, a sum partition is used:  $e_2 = x_1 + x_2$ , where in the classical case the  $x_i \in \mathbb{F}_q^{k+\ell}$  have weight  $v/2 + \varepsilon$ . This is chosen such that  $\varepsilon$  positions of their supports are overlapping and cancel out. Let us first introduce the number of ways we can write  $e_2 = x_1 + x_2$ , i.e., the *number of representations*. For this purpose, one considers a fixed  $e_2$  of weight  $v$  and computes the number of  $x_1$  of weight  $v'$ , such that  $e_2 - x_1$  is of weight  $v'$ . Let us denote this number of representations by  $r$  and  $u = \log_q(r)$ .

Let us now recall how one performs a representation merge: given two lists  $\mathcal{L}_1, \mathcal{L}_2$  containing  $x_i$  of weight  $v'$ , we add  $x = x_1 + x_2$  to the resulting list  $\mathcal{L}$ , whenever  $x$  attains a target weight  $v$  and  $x A_2^\top = t$ , on the first  $u$  positions, for either  $t = s_2$ , the target syndrome or  $t = 0$ , the zero vector. As for any  $x \in \mathcal{L}$  there are  $r$  representations  $(x_1, x_2)$ , which all lead to the same  $x$ , by checking on  $u$  positions, it is guaranteed that one representation for each possible  $x$  survives the merge with high probability. In general, a representation merge of two lists  $\mathcal{L}_1, \mathcal{L}_2$  of asymptotic size  $\Lambda$  on  $u$  positions costs asymptotically

$$2^{n \max\{\Lambda, 2\Lambda - U\}}.$$

After the representation merge, one performs a filtering step, which removes vectors which are not well-formed, e.g., do not achieve a given weight constraint. Further steps can then utilize this smaller list.

In the following, we denote by BJMM( $a$ ) an algorithm that starts with a concatenation merge followed by  $a$  representation merges, since the optimal number of levels  $a$  might change depending on the parameters.

For a BJMM algorithm with  $a$  levels, we denote by  $\Sigma_i$  the size of the search space, by  $V_i = \lim_{n \rightarrow \infty} \frac{v_i(n)}{n}$ , with  $V_0 = V$ , the weight of the vectors, by  $E_i = \lim_{n \rightarrow \infty} \frac{\varepsilon_i(n)}{n}$  the number of overlaps on level  $i \in \{0, \dots, a\}$ , starting from  $a$ . With  $U_i = Q \cdot \lim_{n \rightarrow \infty} \frac{u_i(n)}{n}$  we denote the number of positions on which we merge.

**Theorem 4.** *The enumeration cost of the BJMM(2) algorithm is given by*

$$\max\{\frac{\Sigma_2}{2}, \Sigma_2 - U_1, 2\Sigma_2 - U_1 - U_0, 2\Sigma_1 - U_0 - QL\}, \quad (1)$$

where we can optimize  $V_i, E_i$  under the constraints that

$$V_i = \frac{V_{i-1}}{2} + E_i, \text{ and } \Sigma_i = (R + L)h\left(\frac{V_i}{R+L}\right) + ZV_i,$$

$$U_i = V_i + (R + L - V_i)h\left(\frac{E_{i+1}}{R+L-V_i}\right) + ZE_{i+1}.$$

This algorithm can be used with any number of levels  $a$ , and the cost of a restricted BJMM( $a$ ) algorithm follows straightforwardly. Furthermore, for the subsequent modifications, we always refer to the cost (1), which can be computed using only the sizes of the search spaces and the number of representations.

In the following, we present some new algorithms derived from BJMM to solve the RSDP. For small choices of  $z$ , we can take advantage of the structure of  $\mathbb{E}$ . Following the idea

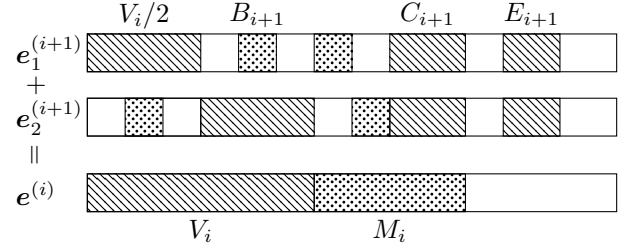


Fig. 1. Counting the number of representations for level  $i$ .

of [3], we add a few elements, denoted by  $\mathbb{E}_+$ , from  $\{\alpha + \beta \mid \alpha, \beta \in \mathbb{E}\}$  to the restricted set  $\mathbb{E}$ , to have more representations. Since we added new elements to the search space, in the final representation merge these elements in  $\mathbb{E}_+$  need to add up to elements in  $\mathbb{E}_0$ . We call those algorithms BJMM( $a$ ) $_+$ , to denote also the number of levels  $a$ .

#### A. Case $z = 2$

We generalize the classical BJMM-like approach by allowing  $1 + 1 = 2$  (and  $-1 + (-1) = -2$ ) in intermediate lists. We call this the BJMM(2) $_+$  algorithm. Thus, in this case we have  $\mathbb{E} = \{\pm 1\}$  and  $\mathbb{E}_+ = \{\pm 2\}$ . This changes the number of representations. In order to construct the intermediate lists using representation merge, we have the usual  $v_i$  entries in  $\{\pm 1\}$  and we also require  $m_i$  to denote the number of  $\pm 2$ 's on level  $i$ . Then, the number of well-formed vectors on level  $i$  is given by  $\binom{k+\ell}{v_i, m_i} 2^{v_i + m_i}$ .

The number of representations of  $e^{(i)} = e_1^{(i+1)} + e_2^{(i+1)}$  on level  $i$  is counted as per Figure 1. For this it is enough to count the number of  $e_1^{(i+1)}$ . There are  $\binom{v_i}{v_i/2}$  ways of splitting the support of the elements in  $\mathbb{E}$ , without choosing the entries. Out of the chosen  $v_i/2$  we chose  $b_{i+1}$  positions, that overlap with  $\pm 2$ 's in  $e_2^{(i+1)}$  and also in the  $v_i/2$  non-chosen positions we choose  $b_{i+1}$  many positions to be  $\pm 2$ . For this we have  $\binom{v_i/2}{b_{i+1}}^2$  possibilities. Out of the  $m_i$  many  $\pm 2$ 's on level  $i$ ,  $c_{i+1}$  are constructed as  $\pm(1 + 1)$ . The remaining  $(m_i - c_{i+1})$  many  $\pm 2$ 's are obtained by support splitting. This results in  $\binom{m_i}{(m_i - c_{i+1})/2, c_{i+1}}$ . Finally, one can choose  $\varepsilon_{i+1}$  out of the  $k + \ell - m_i - v_i$  zero-positions.

While this enlarges the number of well-formed vectors on the intermediate levels, the number of representations is also increased. Let  $M_i = \lim_{n \rightarrow \infty} \frac{m_i(n)}{n}$ ,  $B_i = \lim_{n \rightarrow \infty} \frac{b_i(n)}{n}$ ,  $C_i = \lim_{n \rightarrow \infty} \frac{c_i(n)}{n}$ . Then, the following corollary holds.

**Corollary 5.** *The exponent of the enumeration cost of the BJMM(2) $_+$  algorithm is calculated according to (1). Using*

$$V_i = \frac{V_{i-1}}{2} + E_i + C_i \text{ and } M_i = \frac{M_{i-1} - C_i}{2} + B_i,$$

results in

$$\Sigma_i = (R + L)g\left(\frac{V_i}{R+L}, \frac{M_i}{R+L}\right) + Z(V_i + M_i),$$

$$U_i = V_i \left(1 + h\left(\frac{2B_{i+1}}{V_i}\right)\right) + M_i g\left(\frac{C_{i+1}}{M_i}, \frac{M_i - C_{i+1}}{2M_i}\right)$$

$$+ (R + L - V_i - M_i)h\left(\frac{E_{i+1}}{R+L-V_i-M_i}\right) + ZE_{i+1}.$$

Figure 2 shows the curve of the complexity coefficient  $F(R, W)$  for  $R = 0.5$ ,  $q = 157$  and  $z = 2$ . Classical ISD

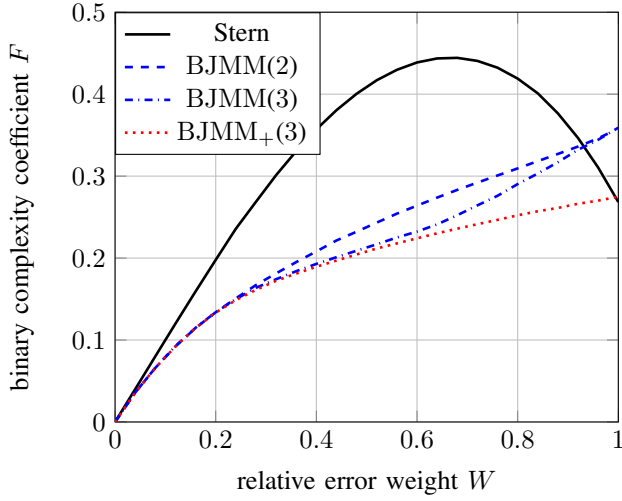


Fig. 2. Comparison of the complexity coefficients  $F(R, W)$  for restricted Stern/Dumer, the general adaption of BJMM and the generalization given in Corollary 5 using  $q = 157$ ,  $z = 2$  and  $R = 0.5$ .

algorithms are usually compared by going through all rates and fixing the weight as large as possible under the uniqueness condition, as this results in the hardest instances. However, for the new RSDP, this is not true in general. Thus, we chose to fix  $R = 0.5$  and go through all weights  $W \in [0, 1]$ , as they all allow for unique decoding. It can be observed that the adapted BJMM algorithm improves significantly over restricted Stern/Dumer for medium error weights. While for the classical SDP two representation levels give the best performance [4], here, three representation layers were found to be optimal. The generalization given in Corollary 5 provides a further improvement for increasing error weights. It was observed that the number of elements from  $\mathbb{E}_+$  is optimized to approximately 0 in the base lists. Hence, one can start with restricted base lists and not lose a noticeable amount of performance.

In [1], the case of  $z = 2$  is considered with the particular choice of  $W = 1$ . As can be seen from Figure 2, in this weight regime, the approach of Corollary 5 does not offer any improvement over Stern/Dumer. For such instances, it is advantageous to shift the error vector  $e$  to  $\tilde{e} = (e + (1, \dots, 1))/2 \in \{0, 1\}^n$ , which is done by computing  $\tilde{s} = (s + (1, \dots, 1)\mathbf{H}^T)/2$ , see e.g. [5]. The resulting error weight is approximately  $\tilde{w} = n/2$ . In order to solve the transformed instance, we follow the BCJ approach [3], i.e., increase the number of representations by allowing  $-1$ 's in intermediate lists, which have to be added to the base lists. In the level  $i$ , for a  $e^{(i)} \in \{0, \pm 1\}^{(k+\ell)}$  with  $v_i$  1's and  $m_i$   $-1$ 's we write  $e^{(i)} = e_1^{(i+1)} + e_2^{(i+1)}$  as in Figure 1, with  $b_{i+1} = c_{i+1} = 0$  and outside of the support we choose  $\varepsilon_{i+1}$  1's that cancel with  $-1$  and  $\varepsilon_{i+1}$   $-1$ 's that cancel with 1's.

**Corollary 6.** The exponent of the shifted BCJ(2) algorithm is calculated according to (1) with

$$\begin{aligned} V_i &= \frac{V_{i-1}}{2} + E_i \text{ and } M_i = \frac{M_{i-1}}{2} + E_i, \\ \Sigma_i &= (R + L) \cdot g\left(\frac{V_i}{R + L}, \frac{M_i}{R + L}\right), \\ U_i &= M_i + V_i \\ &+ (R + L - V_i - M_i)h\left(\frac{2E_{i+1}}{R + L - V_i - M_i}\right) + 2E_{i+1}. \end{aligned}$$

#### B. Case $z = 4$

In this case, we have  $\mathbb{E} = \{\pm 1, \pm g\}$  and define  $\mathbb{E}_+ = \{\pm(g+1), \pm(g-1)\}$ . Following the approach for  $z = 2$ , we obtain the same number of possibilities for choosing the supports of  $e_1^{(i+1)}$  and  $e_2^{(i+1)}$ . There are, however, more possibilities for picking the values in the chosen positions: there are two possibilities for obtaining any  $e \in \mathbb{E}$  as the sum of  $\alpha \in \mathbb{E}$  and  $\beta \in \mathbb{E}_+$  and two possibilities for obtaining  $e \in \mathbb{E}_+$  as the sum of two elements in  $\mathbb{E}$ . This increases the number of representations for level  $i$  overall by factor of  $2^{2(b_{i+1}+c_{i+1})}$  compared to the number computed for  $z = 2$ . Hence, we obtain the same  $V_i, M_i, \Sigma_i$  as in Corollary 5 and the new number of representations is given by

$$\begin{aligned} U_i &= V_i \left(1 + h\left(\frac{2B_{i+1}}{V_i}\right)\right) + M_i g\left(\frac{C_{i+1}}{M_i}, \frac{M_i - C_{i+1}}{2M_i}\right) \\ &+ (R + L - V_i - M_i)h\left(\frac{E_{i+1}}{R+L-V_i-M_i}\right) + ZE_{i+1} \\ &+ 2(B_{i+1} + C_{i+1}). \end{aligned}$$

Figure 3 shows the curve of the complexity coefficient  $F(R, W)$  for  $R = 0.5$ ,  $q = 157$  and  $z = 4$ . Again, BJMM improves over Stern for medium error weights. The generalization using  $\mathbb{E}_+$  gives a further speedup for increased weights.

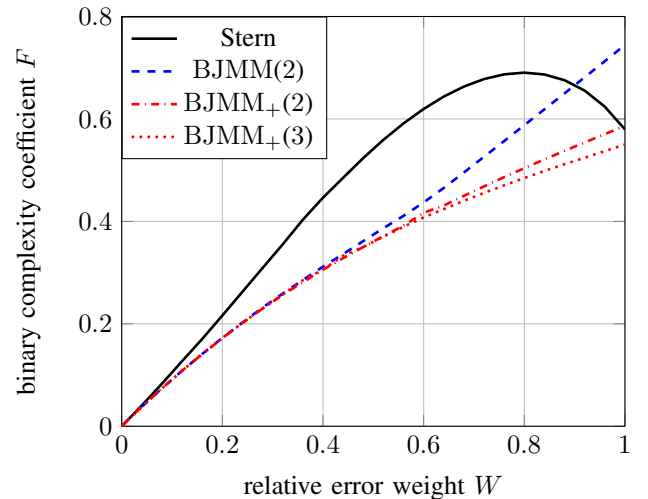


Fig. 3. Comparison of the complexity coefficients  $F(R, W)$  for restricted Stern/Dumer, the general adaption of BJMM and the proposed generalization using  $q = 157$ ,  $z = 4$  and  $R = 0.5$ .

### C. Case $z = 6$

In this case, we have  $\mathbb{E} = \{\pm 1, \pm g, \pm(g-1)\}$ . Note that  $\mathbb{E}$  already possesses additive structure: any element  $e \in \mathbb{E}$  can be obtained as  $e = e_1 + e_2 = e_2 + e_1$  with  $e_1, e_2 \in \mathbb{E}, e_1 \neq e_2$ . This allows setting  $\mathbb{E}_+ = \mathbb{E}$ . Thus, using again Figure 1, there are  $\binom{v_i}{2b_{i+1}, v_i/2-b_{i+1}} 2^{2b_{i+1}}$  representations due to the entries of  $e_1^{(i+1)}$  within the  $v_i$  positions of  $e^{(i)}$  in  $\mathbb{E}$ . The remaining factors are calculated as before, setting  $m_i = c_i = 0$ .

**Corollary 7.** *The exponent of the enumeration cost of the BJMM(2)<sub>+</sub> algorithm is calculated according to (1) using*

$$\begin{aligned} V_i &= \frac{V_{i-1}}{2} + E_i + B_i, \\ \Sigma_i &= (R + L)h\left(\frac{V_i}{R+L}\right) + ZV_i, \\ U_i &= V_i g\left(\frac{2B_{i+1}}{V_i}, \frac{V_i/2-B_{i+1}}{V_i}\right) + 2B_{i+1} \\ &\quad + (R + L - V_i)h\left(\frac{E_{i+1}}{R+L-V_i}\right) + ZE_{i+1}. \end{aligned}$$

Figure 4 shows the curve of the complexity coefficient  $F(R, W)$  for  $R = 0.5$ ,  $q = 157$  and  $z = 6$ . Using the additive structure of  $\mathbb{E}$  as proposed in Corollary 7 enables a remarkable speedup over Stern/Dumer and the basic BJMM adaption. Experiments, for which we allowed elements from  $\{\pm(g+1), \pm(2g-1), \pm(g-2)\}$  in intermediate lists, did not yield further performance improvements.

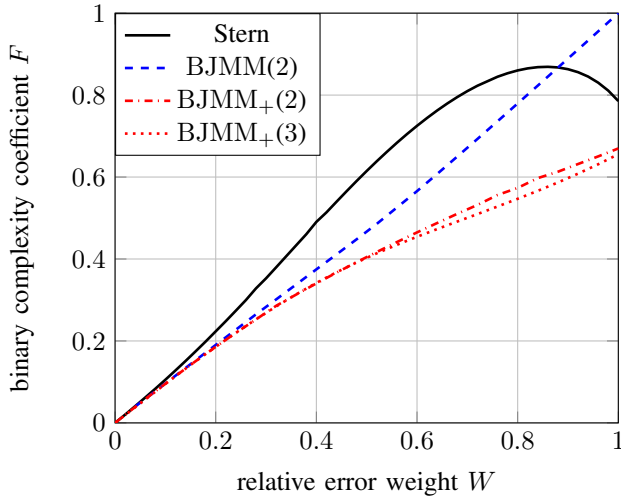


Fig. 4. Comparison of the complexity coefficients  $F(R, W)$  for restricted Stern/Dumer, the general adaption of BJMM and the generalization given in Corollary 7 using  $q = 157$ ,  $z = 6$  and  $R = 0.5$ .

### D. Security Level Update for Instances from Literature

In the following, we apply the presented algorithms to the parameters proposed in [1], [2], [9], [18]. We did not perform a rigorous finite regime analysis, since already approximating the security level as  $2^{F(R, W) \cdot n}$  shows a considerable reduction, compared to the original analysis. The obtained results are summarized in Table I. Python code for reproducing the work factors and the parameters of the decoders is publicly available at [github.com/sebastianbitzer/rest-dec](https://github.com/sebastianbitzer/rest-dec).

TABLE I  
ESTIMATES OF THE WORK FACTORS OF THE PROPOSED ALGORITHMS  
COMPARED TO CLAIMED SECURITY LEVELS FOR INSTANCES FROM  
LITERATURE.

	$z$	$q$	$n$	$R$	$W$	claim (bit)	this work (bit)	
[2]	2	16 381	400	0.75	0.16	128	BJMM(2) <sub>+</sub>	69
[2]	2	16 381	500	0.75	0.13	128	BJMM(2) <sub>+</sub>	76
[2]	2	16 381	400	0.80	0.14	128	BJMM(2) <sub>+</sub>	69
[2]	2	32 749	500	0.75	0.13	128	BJMM(2) <sub>+</sub>	76
[2]	2	32 749	600	0.66	0.14	128	BJMM(2) <sub>+</sub>	82
[1]	2	29	167	0.79	1.00	87	shifted BCJ(3)	48
[1]	2	31	256	0.80	1.00	128	shifted BCJ(3)	74
[18]	4	109	270	0.50	0.34	125	BJMM(2) <sub>+</sub>	75
[18]	4	157	312	0.50	0.34	144	BJMM(2) <sub>+</sub>	85
[18]	4	197	384	0.50	0.34	177	BJMM(2) <sub>+</sub>	103
[9]	4	137	272	0.20	0.60	88	BJMM(2) <sub>+</sub>	45
[9]	4	157	312	0.20	0.60	101	BJMM(2) <sub>+</sub>	51
[9]	4	173	344	0.20	0.60	111	BJMM(2) <sub>+</sub>	56
[9]	4	193	384	0.20	0.60	124	BJMM(2) <sub>+</sub>	62
[18]	6	139	276	0.02	0.60	89	BJMM(3) <sub>+</sub>	50
[18]	6	157	312	0.20	0.60	101	BJMM(3) <sub>+</sub>	55
[18]	6	193	384	0.20	0.60	124	BJMM(3) <sub>+</sub>	67

### E. Arbitrary $z$

While the presented attacks focused on error sets of size  $z \in \{2, 4, 6\}$ , the proposed solvers can also be generalized to larger values of  $z$ . Such a generalization can utilize any additive structure of  $\mathbb{E}_0$ . The concrete structure of  $\mathbb{E}_0$  depends on the factorization of  $x^z - 1$  in  $\mathbb{F}_q$ . As the factorization cannot be given in general, any proposed  $z$  should be checked independently. As we have seen, it can be beneficial to allow elements from  $\mathbb{E}_+$  to increase the number of representations. Finally, the possibility of transforming the problem by shifting the error vector has to be taken into account.

## V. CONCLUSION

In this paper, we studied the complexity of the restricted syndrome decoding problem, which has recently gained attention in code-based cryptography. To this end, we adapted the representation technique, which is utilized in the fastest known solvers for syndrome decoding problems to the new setting. In particular, small choices of the restriction cardinality  $z$  were considered, for which we provided novel tailored solvers which are inspired by [3]. This leads to a drastic decrease of the respective security levels. Nevertheless, we believe that the restricted syndrome decoding problem is a promising underlying problem for cryptographic applications. In contrast to previous proposals, we would like to advocate the use of larger values of  $z$ .

## ACKNOWLEDGEMENTS

Sebastian Bitzer acknowledges the financial support by the Federal Ministry of Education and Research of Germany in the programme of “Souverän. Digital. Vernetzt.”. Joint project 6G-life, project identification number: 16KISK002. Violetta Weger is supported by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement no. 899987.

## REFERENCES

- [1] Marco Baldi, Massimo Battaglioni, Franco Chiaraluce, Anna-Lena Horlemann-Trautmann, Edoardo Persichetti, Paolo Santini, and Violetta Weger. A new path to code-based signatures via identification schemes with restricted errors. *arXiv preprint arXiv:2008.06403*, 2020.
- [2] Marco Baldi, Franco Chiaraluce, and Paolo Santini. Code-based signatures without trapdoors through restricted vectors. *Cryptology ePrint Archive*, 2021.
- [3] Anja Becker, Jean-Sébastien Coron, and Antoine Joux. Improved generic algorithms for hard knapsacks. In *Advances in Cryptology – EUROCRYPT 2011*, pages 364–385. Springer, 2011.
- [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 *Advances in Cryptology – EUROCRYPT 2012*, pages 520–536. Springer, 2012.
- [5] Rémi Bricout, André Chailloux, Thomas Debris-Alazard, and Matthieu Lequesne. Ternary syndrome decoding with large weight. In *Selected Areas in Cryptography – SAC 2019*, pages 437–466. Springer, 2020.
- [6] Kevin Carrier, Thomas Debris-Alazard, Charles Meyer-Hilfiger, and Jean-Pierre Tillich. Statistical decoding 2.0: Reducing decoding to LPN. *arXiv preprint arXiv:2208.02201*, 2022.
- [7] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. Wave: A new family of trapdoor one-way preimage sampleable functions based on codes. In *Advances in Cryptology – ASIACRYPT 2019*, pages 21–51. Springer, 2019.
- [8] Il'ya I. Dumer. Two decoding algorithms for linear codes. *Problemy Peredachi Informatsii*, 25(1):24–32, 1989.
- [9] Jürgen Freudenberger and Johann-Philipp Thiers. A new class of  $q$ -ary codes for the McEliece cryptosystem. *Cryptography*, 5(1):11, 2021.
- [10] Cheikh Thiécoumba Gueye, Jean Belo Klamti, and Shoichi Hirose. Generalization of BJMM-ISD using May-Ozerov nearest neighbor algorithm over an arbitrary finite field  $\mathbb{F}_q$ . In *International Conference on Codes, Cryptology, and Information Security*, pages 96–109. Springer, 2017.
- [11] Nick Howgrave-Graham and Antoine Joux. New generic algorithms for hard knapsacks. In *Advances in Cryptology – EUROCRYPT 2010*, pages 235–256. Springer, 2010.
- [12] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.
- [13] Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in  $\mathcal{O}(2^{0.054n})$ . In *Advances in Cryptology – ASIACRYPT 2011*, pages 107–124. Springer, 2011.
- [14] Alexander Meurer. *A coding-theoretic approach to cryptanalysis*. PhD thesis, Ruhr-Universität Bochum, 2013.
- [15] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8(5):5–9, 1962.
- [16] Richard Schroeppel and Adi Shamir. A  $t = \mathcal{O}(2^{n/2})$ ,  $s = \mathcal{O}(2^{n/4})$  algorithm for certain NP-complete problems. *SIAM journal on Computing*, 10(3):456–464, 1981.
- [17] Jacques Stern. A method for finding codewords of small weight. In *International Colloquium on Coding Theory and Applications*, pages 106–113. Springer, 1988.
- [18] Johann-Philipp Thiers and Jürgen Freudenberger. Generalized concatenated codes over Gaussian and Eisenstein integers for code-based cryptography. *Cryptography*, 5(4):33, 2021.
- [19] Violetta Weger, Niklas Gassner, and Joachim Rosenthal. A survey on code-based cryptography. *arXiv preprint arXiv:2201.07119*, 2022.
- [20] Violetta Weger, Karan Khathuria, Anna-Lena Horlemann, Massimo Battaglioni, Paolo Santini, and Edoardo Persichetti. On the hardness of the Lee syndrome decoding problem. *Advances in Mathematics of Communications*, 2022.