McEliece in the world of Escher

Danilo Gligoroski[1] and Simona Samardjiska[1,2] and Håkon Jacobsen[1] and Sergey Bezzateev[3]

[1] Department of Telematics, Norwegian University of Science and Technology (NTNU), Trondheim, NORWAY,
{danilog, simonas, hakoja}@item.ntnu.no

[2] "Ss Cyril and Methodius" University, Faculty of Computer Science and Engineering (FINKI), Skopje, MACEDONIA
simona.samardjiska@finki.ukim.mk

[3] Saint Petersburg State University of Aerospace Instrumentation, Saint Petersburg, RUSSIA,
bsv@aanet.ru

Abstract. We present a new family of linear binary codes of length n and dimension k accompanied with a fast list decoding algorithm that can correct up to $\frac{n}{2}$ errors in a bounded channel with an error density ρ. The decisional problem of decoding random codes using these generalized error sets is NP-complete. Next we use the properties of these codes to design both an encryption scheme and a signature scheme. Although in the open literature there have been several proposals how to produce digital signatures from the McEliece public key scheme, as far as we know, this is the first public key scheme based on codes where signatures are produced in a straightforward manner from the decryption procedure of the scheme. The security analysis of our scheme have four parts:

1. An extensive list of attacks using the Information Set Decoding techniques adopted for our codes;
2. An analysis of the cost of a distinguishing attack based on rank attacks on the generator matrix of the code or on its dual code;
3. An analysis of the cost of cheap distinguishing attacks on the generator matrix of the code or on its dual code that have expensive list-decoding properties;
4. we interpret our scheme as multivariate quadratic system and discuss difficulties of solving that system using algebraic approaches such as Gröbner bases.

Based on this security analysis we suggest some concrete parameters for the security levels in the range of $2^{80}-2^{128}$. An additional feature of the decryption process is that it admits massive and trivial parallelization that could potentially make our scheme in hardware as fast as the symmetric crypto primitives.

Keywords: Public Key, Cryptography, McEliece PKC, Error Correcting Codes, List Decoding

Table of Contents

1 Introduction

The McEliece public key scheme [36] was published two years after the seminal paper of Diffie and Hellman [17] and was the first scheme based on the theory of error-correcting codes and the NP-hardness of the problem of decoding random linear codes. The original scheme used binary Goppa codes with parameters $[n, k, 2t + 1] = [1024, 524, 101]$ for a security level of 2^{80} operations and a public key size of around 32 kB. This was probably one of the main reasons why the scheme was not widely used in practice, despite the fact that encryption and decryption were much faster than in RSA [42].

Still, the McEliece PKC has received a considerable amount of cryptanalytic effort, and has upheld remarkably well. Apart from an update of the original parameters due to improvements in Information Set Decoding (ISD) techniques [30,44,9,20,12,35,7], the main design remains sound.

1.1 Related Work

Three research directions in code-based cryptography are related to the work in this paper: 1. Use of alternative codes instead of binary Goppa codes; 2. Use of list-decoding techniques in code-based cryptography; 3. Design of code-based digital signature schemes.

Alternative codes: Soon after McEliece published his scheme based on binary Goppa codes, several alternatives using different codes were proposed. For example in [39] Niederreiter proposed to use generalized Reed-Solomon codes, Gabidulin et al., in [21] proposed the use of rank codes, Sidelnikov [43], proposed the use of binary Reed-Muller codes, Janwa and Moreno in [26] used algebraic-geometric codes, Gaborit in [22] used cyclicity and quasi-cyclicity of BCH codes in order to shorten the length of the public key, Monico et al., [38], proposed replacement of the binary Goppa codes with Low Density Parity Check (LDPC) codes, then Baldi et al., in [5] extended that idea to use Quasi-Cyclic LDPC codes, and Misoczki et al., [37], instead of LDPC codes proposed the use od Middle Density Parity Check (MDPC) codes.

List-decoding techniques: The idea of list decoding was present in the literature from the late 50's [18,48], but an efficient algorithm with polynomial run-time was published four decades later by Sudan [45] and subsequently was significantly improved by Guruswami and Sudan in [23]. In code-based cryptography, the use of list decoding techniques came later in the works of Bernstein, Lange and Peters [10,11,13].

Code-based digital signature schemes: Early attempts [47,32,25,3,28] to design a code-based signature scheme proved to be unsuccessful and were broken [2,1,46,49,14]. Similarly, some newer schemes such as [24,33] have been broken in [29,40]. There is an intrinsic difficulty in designing a signature scheme from the McEliece scheme. The reason is that for the signature part someone needs to decode a random syndrome which is generally a hard problem. In 2001 Courtois, Finiasz and Sendrier proposed a signature scheme [16] that so far has resisted cryptanalytic attacks. However, compared to the signatures schemes based on number theory or discrete logarithm problem on elliptical curves, it is not very practical: It has a big public key, the speed of producing signatures is much slower and is not scalable for security levels beyond the 2^{128} range.

1.2 Our Contribution

All known code-based PKC schemes are based on codes where no structure is imposed on the error vectors except for the requirement of having Hamming weight[4] less than, or equal to a certain value t. When modeling a noisy channel this is a natural approach, however, the cryptographic setting is an artificial one. In this paper we novate the use of the noisy channel with a channel where the sender has full control over the "noise" and can produce error vectors with a significantly different pattern than in the classical case. We call a collection of such error vectors an *error set*. We define two important characteristics of these error sets: *density* and *granulation*.

In the classical case, the set of all syndromes is partially covered by Hamming spheres and there is a unique decoding if the norm of the error is less than $t = \frac{d}{2}$, where d is the minimum distance of the code. On the other hand, in our approach using error sets, we can cover almost the complete set of all syndromes (except a negligible portion) with a tessellation around the code words. However, we do not have a unique decoding. Intuitively this covering can be represented in a form of an artistic Escher's tessellation [5]. A graphical presentation of the

[4] Or other norms such as the Rank norm used in Gabidulin codes [21].

[5] M. C. Escher (1898 - 1972), Dutch graphical artist. Known for his drawings of impossible, self-referential constructions.

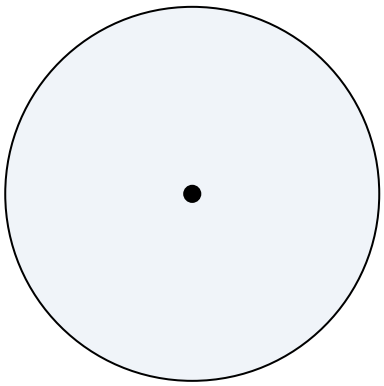


Fig. 1. A classical modeling of an error set around a code word - The Hamming sphere.

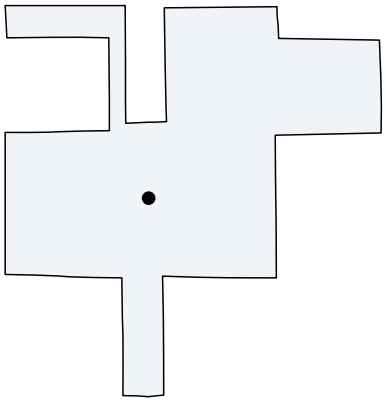


Fig. 2. In our approach an error set around a code word can be an arbitrary set.

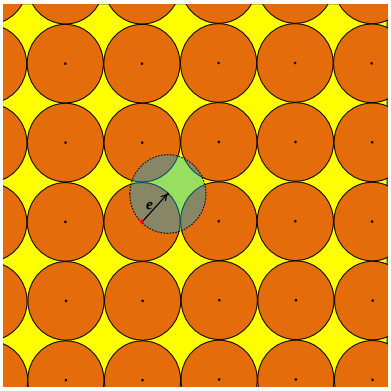


Fig. 3. A classical modeling of an error set around a code word with the Hamming sphere. If the error is less than the $\frac{d}{2}$ where d is the minimal distance of the code, we have a unique decoding.

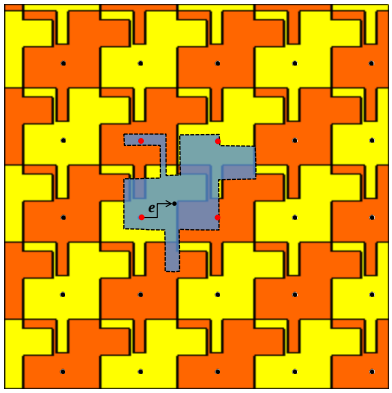


Fig. 4. An artistic visualization of our idea with an arbitrary error set around the codewords. The tessellation of the plane with these error sets is similar to Escher's tessellations. There is no unique decoding in this case.

conceptual differences between the classical approach of using error sets that form a Hamming sphere around the codewords and our approach, using arbitrary error sets, is given in figures Fig. 1, Fig. 2, Fig. 3 and Fig. 4.

We introduce a new family of binary linear codes that with overwhelming probability can decode a given error set. The generator matrix G of these codes has a stepwise random block structure, which enables us to define an efficient list decoding algorithm.

Having introduced the new codes, we derive both encryption and signature schemes that follow the basic structure of the McEliece scheme: $G_{\text{pub}} = SGP$. Like in other code-based schemes, the security of our schemes depends both on the hardness of decoding random syndromes and on the hardness of recovering the underlying structure of the code. We show that the first is tightly connected to the decoding of random syndromes in the Hamming metric. In particular, the related decisional problem of decoding random codes using our generalized error sets is NP-complete. Then we provide an analogue to the Information Set Decoding techniques for our error sets. For the second assumption, we note that the particular structure of G (and its dual parity check matrix H) can be a source of weakness. Thus, we make a careful trade off between the size of the stepwise structure present in G (and H), the size of the internal blocks of G (and H), the efficiency of the scheme and the security of the scheme.

As a concrete example of our construction we consider the following error set $E = \{00, 01, 10\}$. The error vectors $\mathbf{e} \in \mathbb{F}_2^n$ will be constructed as the concatenation of $m = \frac{n}{2}$ randomly drawn elements from the error set E. That is, $\mathbf{e} = \mathbf{e}_1 \| \mathbf{e}_2 \| \dots \| \mathbf{e}_m$, with each $\mathbf{e}_i \in E$. Like in the McEliece scheme we need our error sets to

be invariant under the permutation P. Therefore we can not choose P from the set of all $n \times n$ permutation matrices. Instead, we use block permutation matrices that permute the m substrings of $\mathbf{e}$.

A unique characteristic of our code-based scheme is that the encryption scheme can be turned into a signature scheme directly using the decoding (decryption) algorithm. In other code-based signature schemes, like [16], the probability of finding a decodable syndrome is relatively small. To remedy this, the strategy is to introduce a counter and produce syndromes as $Syndrome = Hash(Doc, Counter)$ until a decodable one is found. In our scheme, with high probability, we can apply the decoding directly on the value $Syndrome = Hash(Doc)$.

2 Notation and Preliminaries

Throughout the paper, we will denote by $\mathcal{C} \subseteq \mathbb{F}_2^n$ a binary (n, k) code of length n and dimension k. We will denote the generator matrix of the code by G, and $wt(\mathbf{x})$ will denote the Hamming weight of the word $\mathbf{x}$.

Unlike the standard approach in code-based cryptography that relies on the Hamming metric and unique decoding, we will use a different characterization parameter and list decoding technique that enables correct decoding with overwhelming probability. We will need some new notions, to our knowledge, previously not used in code-based cryptography.

Definition 1. *Let ℓ be a positive integer and let $S \subset \mathbb{F}_2[x_1, x_2, \ldots, x_\ell]$ be a set of multivariate polynomials. We say that E_ℓ is an* error set *if it is the kernel of S.*

$$E_\ell = \mathrm{Ker}(S) = \{\mathbf{e} \in \mathbb{F}_2^\ell \mid f(\mathbf{e}) = 0, \forall f \in S\}. \tag{1}$$

We define the density *of the error set E_ℓ to be $\rho_\ell = D(E_\ell) = |E_\ell|^{1/\ell}$. We will refer to the integer $\ell > 0$ as the* granulation *of E_ℓ (when it is clear from context, we will drop the subscript ℓ).*

Immediately we have the following proposition:

Proposition 1. *1. Let $E_{\ell_1} \subseteq \mathbb{F}_2^{\ell_1}$, $E_{\ell_2} \subseteq \mathbb{F}_2^{\ell_2}$, for some integers $\ell_1, \ell_2 > 0$. Let $D(E_{\ell_1}) = D(E_{\ell_2}) = \rho$. Then $D(E_{\ell_1} \times E_{\ell_2}) = \rho$.*
2. Let $E_{\ell,1}, E_{\ell,2}, \ldots, E_{\ell,m} \subseteq \mathbb{F}_2^\ell$, $\ell > 0$, and $D(E_{\ell,1}) = D(E_{\ell,2}) = \cdots = D(E_{\ell,m}) = \rho$. Then $D(E_{\ell,1} \times E_{\ell,2} \times \cdots \times E_{\ell,m}) = \rho$.

Proof. 1. $D(E_{l_1} \times E_{l_2}) = |E_{l_1} \times E_{l_2}|^{1/(l_1+l_2)} = (|E_{l_1}| \cdot |E_{l_2}|)^{1/(l_1+l_2)} = (\rho^{l_1} \cdot \rho^{l_2})^{1/(l_1+l_2)} = \rho$.
2. Follows directly from 1.

□

Example 1.

1. Let $E_2 = \{\mathbf{x} \in \mathbb{F}_2^2 \mid wt(\mathbf{x}) < 2\} = \{(0,0), (0,1), (1,0)\}$. Then $D(E_2) = |E_2|^{1/2} = 3^{1/2}$, and also $D(E_2^2) = |E_2^2|^{1/4} = 9^{1/4} = 3^{1/2}$ as well as $D(E_2^m) = 3^{1/2}$ for any positive integer m.
2. Let $E_{4,1} = \{\mathbf{x} \in \mathbb{F}_2^4 \mid 2 \leq wt(\mathbf{x}) \leq 3\}$. Then $D(E_{4,1}) = (\sum_{i=2}^{3} \binom{4}{i})^{1/4} = 10^{1/4}$, and also $D(E_{4,1}^m) = 10^{1/4}$ for any positive integer m. Note that the set $E_{4,2} = \{\mathbf{x} \in \mathbb{F}_2^4 \mid wt(\mathbf{x}) \leq 2\} \setminus \{(0,0,0,0)\}$ also has density $D(E_{4,2}) = 10^{1/4}$.
3. Let $E_4 = \{(0,1,0,0), (0,0,0,1), (0,1,0,1), (1,0,0,1), (0,0,1,0), (0,1,1,0), (1,0,1,0), (1,1,1,0), (0,1,1,1), (1,1,1,1)\}$. The values of E_4 are chosen without any particular rule in mind. Then $D(E_4) = |E_4|^{1/4} = 10^{1/4}$ as well as $D(E_4^m) = 10^{1/4}$ for any positive integer m.

We will be interested in finding codes that can correct errors vectors drawn from $E_n = E_\ell^m = E_\ell \times E_\ell \times \cdots \times E_\ell$ of a given density ρ_ℓ. These error sets differ from the standard error sets usually considered in code-based cryptography. In particular, the error sets usually used are determined by the ability of the code to uniquely decode such errors with respect to some metric, like the Hamming metric or rank metric. While this approach guarantees unique decoding, the size of the error set is restricted to a relatively small number and is given by the well known Hamming bound for (n, k) binary code of minimum distance d: We will be interested in finding codes that can correct error sets $E_n = E_\ell^m = E_\ell \times E_\ell \times \cdots \times E_\ell$ of a given density $\rho_\ell = D(E_\ell) > 2^{(\ell-1)/\ell}$. These error sets differ from the standard error sets usually considered in code-based cryptography. In particular, the error sets usually used are determined by the ability of the code to uniquely decode such errors with respect

to some metric, like the Hamming metric or rank metric. While this approach guarantees unique decoding, the size of the error set is restricted to a relatively small number and is given by the well known Hamming bound for (n, k) binary code of minimum distance d: $k \leq n - \log_2 \left(\sum_{i=0}^{\lfloor \frac{d-1}{2} \rfloor} \binom{n}{i} \right)$.

In this work, instead of unique decoding, we take the approach of *list decoding*, a notion that dates back to the work of Elias [18] and Wozencraft [48] in the 1950's. In list decoding, the decoder is allowed to output a list of possible messages one of which is correct. List decoding can handle a greater number of errors than that allowed by unique decoding. In order for the list decoding to be efficient, the size of the resulting list has to be polynomial in the block length of the code.

Several bounds exist that link the error rates beyond the Hamming bound and rates of codes that can efficiently decode them with overwhelming probability. For example, the list decoding capacity, i.e., the information theoretic limit of list decodability, is given by the optimal trade-off between the code rate and the fraction of errors that can be corrected under list decoding. This bound shows that list decoding can correct twice as many errors as unique decoding, for every rate (see for ex. [23]). Another bound, the Johnson bound [27] gives the radius of a Hamming ball beyond half the minimum distance up to which any code of a given distance can be list decoded using polynomial lists.

Here we derive some bounds for codes that link the density of the error sets to the code rate. First, we recall a simpler variant [4] of the Chernoff bound [15] that gives an estimate on the tail in a binomial distribution.

Lemma 1. (Chernoff bound)[4] *Let X_i, $i = 1, \ldots, N$ be independent binary random variables with $\Pr[X_i = 1] = p$. Then the following bounds are true:*

$$\Pr[\sum_{i=1}^{N} X_i \leq (1-\epsilon)pN] \leq e^{-\epsilon^2 pN/2}, \quad \text{for all } 0 < \epsilon \leq 1, \tag{2}$$

$$\Pr[\sum_{i=1}^{N} X_i \geq (1+\epsilon)pN] \leq e^{-\epsilon^2 pN/(2+\epsilon)}, \quad \text{for all } \epsilon > 1. \tag{3}$$

In essence, the Chernoff bound states that the probability mass is concentrated around the mean pN which is the expected value for $\sum_{i=1}^{N} X_i$.

As a consequence of Lemma 1 we have the following bound that is true for any binary code:

Proposition 2. *Let $\mathcal{C}$ be any binary (n, k) code and $E \subset \mathbb{F}_2^n$ be an error set of density ρ. Let $\mathbf{w}$ be any word of length n, $W_E = \{\mathbf{w} + \mathbf{e} \mid \mathbf{e} \in E\}$ and $\mathcal{C}_{W_E}$ denote the set of codewords in W_E. Then:*

1. *The expected number of codewords in W_E is $\rho^n 2^{k-n}$. The probability that $\mathcal{C}_{W_E}$ is an empty set is estimated by $\Pr[|\mathcal{C}_{W_E}| \leq 1/2] \leq e^{-(\rho^n 2^{k-n+1}-1)^2/(\rho^n 2^{k-n+3})}$.*
2. *Suppose there exists a codeword $\mathbf{c} \in W_E$. Then the expected number of codewords in $W_E \backslash \{\mathbf{c}\}$ is approximately $\rho^n 2^{k-n}$ for large enough n and k. The probability that $\mathcal{C}_{W_E \backslash \{\mathbf{c}\}}$ has another element except $\mathbf{c}$ is estimated by $\Pr[|\mathcal{C}_{W_E \backslash \{\mathbf{c}\}}| \geq 1/2] \leq e^{-(1-\rho^n 2^{k-n+1})^2/2(1+\rho^n 2^{k-n+1})}$.*

Proof. 1. Since $D(E) = \rho$, we have that $|E| = \rho^n$, and thus $|W_E| = \rho^n$. From here it follows that the probability that a random word is in the set W_E is $p = \rho^n/2^n$. We can consider the event that one codeword is in the set W_E as independent from the event that another codeword is in W_E. There are 2^k codewords, so it follows directly that the expected number of codewords in W_E is $\rho^n 2^{k-n}$.
For the second part, let $N = 2^k$ and fix an enumeration $\mathbf{c}_1, \mathbf{c}_2, \ldots \mathbf{c}_N$ of $\mathcal{C}$. By letting the random variables X_i be 1 iff $\mathbf{c}_i \in W_E$ in Lemma 1, and setting $\epsilon = 1 - 1/(2pN)$ in (2), the claim follows.

2. In this situation we again have a sequence of independent events. Now, the number of codewords except c is $2^k - 1$, and the probability that a random word except c is in the set $W_E \setminus \{c\}$ is $p = (\rho^n - 1)/(2^n - 1)$. Now the expected number of codewords in $W_E \setminus \{c\}$ is $(\rho^n - 1)(2^k - 1)/(2^n - 1)$ which can be approximated to $\rho^n 2^{k-n}$ for large enough n and k.
The second part follows directly from Lemma 1, by setting $\epsilon = 1/(2pN) - 1$ in (3).

□

We illustrate the implications of Proposition 2 through some examples.

Example 2.

1. Let $\mathcal{C}$ be a $(1280, 256)$ binary code. The code rate is 0.2. We consider an error set E of density $\rho = 3^{1/2}$. Let $\mathbf{c}$ be a codeword and $\mathbf{w} = \mathbf{c} + \mathbf{e}$ for some $\mathbf{e} \in E$. Then, from Proposition 2 the decoding list of the word $\mathbf{w}$ is of average length $1 + \text{Exp}[|\mathcal{C}_{W_E \setminus \{\mathbf{c}\}}|] = 1.00127$. The probability that there is another element in the list except $\mathbf{c}$ is 0.6. Note that these parameters may be suitable for building an encryption scheme, since we can expect that the list has only one element.
2. Let $\mathcal{C}$ be a $(1208, 256)$ binary code. The code rate is 0.211921. We consider an error set E of density $\rho = 3^{1/2}$. Let $\mathbf{w}$ be a word of length n. Then the decoding list of the word $\mathbf{w}$ is of average length 39.8733, and the probability that the list is empty is 2^{-28}. Such parameters are suitable for building a signature scheme, since with great confidence we can always expect to have a valid signature. Moreover, the number of valid signatures is relatively small.

Note that it is possible to define an error set with small enough density that does not exceed the Hamming bound. In that case, knowing the minimum distance of a code will give constrains for unique decoding. Our bound does not guarantee unique decoding, but gives a good estimate,even for codes for which we do not know the minimum distance.

Example 3. Let $\mathcal{C}$ be a $(1024, 524)$ binary code of minimum distance $d = 101$. Note that these are the parameters of the original McElliece system. The size of the error set of uniquely decodable errors is $\sum_{i=0}^{50} \binom{1024}{i} \approx 2^{284}$. This error set has density $\rho = 1.21$. Let $\mathbf{c}$ be a codeword, and $\mathbf{w} = \mathbf{c}+\mathbf{e}$ for some $\mathbf{e} \in E$. Then, from Proposition 2 the decoding list of the word $\mathbf{w}$ is of average length $1+2^{-216} \approx 1$. Hence, the list decoding bound from Proposition 2 gives a very good estimate of the size of the decoding list, even though it does not guarantee uniqueness (which for these parameters is the case).

Even more, if we allow errors up to weight 100 (twice as big as for unique decoding), we get an error set of density $\rho = 1.37$. Now the decoding lists will on average have length of $1 + 2^{-32}$, hence we can still expect that there will only be one element in it in most of the cases.

Having introduced the definitions and the basic properties for the generalized error sets, we have the following Theorem:

Theorem 1. *The decisional problem of decoding random codes using the error sets in Def. 1 is NP-complete.*

Proof. If $\mathcal{C}$ is an (n, k) binary linear code and $\mathbf{y}$ a received vector, the general decoding problem asks to find the codeword that most likely was sent. That is, "most-likely" means to find an error vector $\mathbf{e}$ of minimal weight, such that $\mathbf{x} = \mathbf{y} + \mathbf{e}$ is word in the code. Equivalently, if H is the $(n - k) \times n$ parity-check matrix of $\mathcal{C}$, the problem can be stated as asking to find a minimal weight solution $\mathbf{e}_0$ to the equation $\mathbf{s} = \mathbf{e}H^T$, where $\mathbf{s}$ is the *syndrome* $\mathbf{y}H^T$.

The hardness of decoding arbitrary linear codes have long been established based on the decades of efforts trying to solve it. Furthermore, Berlekamp et al. [8], showed that the related decisional problem, COSET WEIGHTS, is NP-complete:

Definition 2 (COSET WEIGHTS).

Input: *A binary $(n - k) \times n$ matrix H, a binary vector $\mathbf{s} \in \mathbb{F}_2^{n-k}$, and a non-negative integer w.*
Output: *YES, if there exists a vector $\mathbf{e}$ of Hamming weight $\leq w$ such that $\mathbf{e}H^T = \mathbf{s}$. NO otherwise.*

We now show that the problem of decoding arbitrary linear codes when using the error sets defined in Def. 1 is NP-complete.

Definition 3 (ERROR SETS).

Input: *A binary $(n - k) \times n$ matrix H, a binary vector $\mathbf{s} \in \mathbb{F}_2^{n-k}$, and a generalized error set $E_\ell \subset \mathbb{F}_2^\ell$.*
Output: *YES, if there exists an error vector $\mathbf{e} \in E_\ell^m$ such that $\mathbf{e}H^T = \mathbf{s}$. NO otherwise.*

Proving ERROR SETS NP-complete amounts to nothing more than noting that the set of errors having Hamming weight $\leq w$ constitutes a generalized error set. That is, COSET WEIGHTS is simply a special case of ERROR SETS by letting $\ell = n$ and

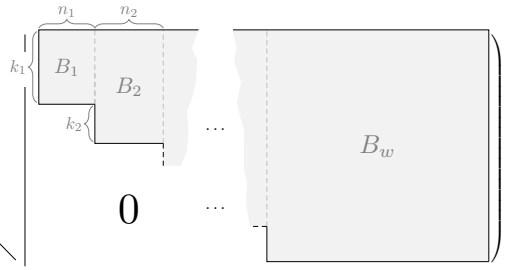
$$E_\ell = \text{Ker}(\{\prod_{i \in I \subseteq [n]} x_i \in \mathbb{F}_2[x_1, \ldots, x_n] \mid |I| > w\}) = \{\mathbf{e} \in \mathbb{F}_2^n \mid wt(\mathbf{e}) \leq w\}, \tag{4}$$

Thus, since a solution to ERROR SETS would be a solution to COSET WEIGHTS, we have proved Thm. 1. □

3 Concrete Code Example

We consider a binary (n, k) code $\mathcal{C}$ with the following generator matrix on standard form:

$$G = \begin{pmatrix} & & & & & & \\ & I_k & & & & & \\ & & & & & & \\ & & & 0 & & & \end{pmatrix}$$



(5)

Each B_i is a random binary matrix of dimension $\sum_{j=1}^{i} k_j \times n_i$, so that $k = k_1 + k_2 + \cdots + k_w$ and $n = k + n_1 + n_2 + \cdots + n_w$.

Let E_ℓ be an error set with density ρ where ℓ divides n and $m = n/\ell$. We describe a general list decoding algorithm (Alg. 1) for the code $\mathcal{C}$, that corrects errors from the set E_ℓ^m.

Algorithm 1 Decoding

Input: A vector $\mathbf{y} \in \mathbb{F}_2^n$, and a generator matrix G of the form (5).

Output: A list $L_w \subset \mathbb{F}_2^k$ of valid decodings of $\mathbf{y}$.

Procedure:
Let $K_i = k_1 + \cdots + k_i$. Represent $\mathbf{x} \in \mathbb{F}_2^k$ as $\mathbf{x} = \mathbf{x}_1 \parallel \mathbf{x}_2 \parallel \cdots \parallel \mathbf{x}_w$ where each $\mathbf{x}_i$ has length k_i. Similarly, for $\mathbf{y} \in \mathbb{F}_2^n$, represent it as $\mathbf{y} = \mathbf{y}_0 \parallel \mathbf{y}_1 \parallel \mathbf{y}_2 \parallel \cdots \parallel \mathbf{y}_w$, where each $\mathbf{y}_i$ has length n_i and $|\mathbf{y}_0| = k$. We further identify $\mathbf{y}_0$ with $\mathbf{y}_0 = \mathbf{y}_0[1] \parallel \mathbf{y}_0[2] \parallel \cdots \parallel \mathbf{y}_0[w]$, where each $\mathbf{y}_0[i]$ is of length k_i.

During decoding, we will maintain the lists $L_1, L_2, \ldots, L_w$ of possible decoding candidates of length K_i.

Step 0: Set a temporary list $T_0 = L_0$ to contain all possible decodings of the first k_1 coordinates of $\mathbf{y}$:

$$T_0 \leftarrow \{\mathbf{x}' = \mathbf{y}_0[1] + \mathbf{e} \mid \mathbf{e} \in E^{k_1/\ell}\}.$$

Step $1 \le i \le w$**:** Perform list-decoding to recover a list of valid decodings:

For each candidate $\mathbf{x}' \in T_{i-1} \subset \mathbb{F}_2^{K_i}$, add to L_i all the candidates for which $\mathbf{x}'B_i + \mathbf{y}_i \in E^{n_i/\ell}$:

$$L_i \leftarrow \{\mathbf{x}' \in T_{i-1} \mid \mathbf{x}'B_i + \mathbf{y}_i \in E^{n_i/\ell}\}. \tag{6}$$

If $i < w$ then create the temporary list T_i of candidates of length K_{i+1} from L_i:

$$T_i \leftarrow \{\mathbf{x}' \parallel (\mathbf{y}_0[i+1] + \mathbf{e}) \mid \mathbf{x}' \in L_i,\ \mathbf{e} \in E^{k_{i+1}/\ell}\}. \tag{7}$$

Return: L_w.

Remark 1. Note that, when testing the validity of a candidate in (6) in Step i of Alg. 1, it is not necessary to consider all errors of length n_i for all candidates in T_{i-1}. Instead, one can incrementally add in more and more constraints (by using more and more columns of B_i) until a candidate either: fails to be a valid decoding, in which case we discard it immediately, or all n_i columns of B_i have been considered. In practice, this strategy will remove most of the bad candidates without considering all errors of length n_i, avoiding much unnecessary computation.

3.1 Small Decoding Example

In this example we consider a binary $(24, 6)$ code generated by:

$$G = \begin{pmatrix} 1 & & & 0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,1\,0\,0\,0\,1\,1\,0 \\ & 1 & & 0\,0\,1\,1\,0\,1\,0\,1\,0\,1\,0\,0\,0\,1\,0\,0\,0\,0 \\ & & 1 & 1\,1\,0\,1\,0\,1\,0\,0\,0\,0\,1\,0\,0\,1\,0\,0\,0\,1 \\ & & & 0\,0\,1\,1\,1\,0\,1\,0\,1\,1\,0\,1\,1\,0\,1\,0\,0\,1 \\ & & & 0\,1\,1\,1\,1\,1\,1\,1\,1\,1\,0\,1\,1 \\ & & & 1\,1\,0\,0\,1\,1 \end{pmatrix}. \tag{8}$$

In particular, we have $k_1 = 4$, $k_2 = 1$, $k_3 = 1$ and $n_1 = n_2 = n_3 = 6$ (ref. (5)). Let $E_\ell = \{00, 01, 10\}$ be our error set, having granulation $\ell = 2$ and density $\rho = 3^{1/2}$. For the following values of the message $\mathbf{x}$ and error vector $\mathbf{e}$, we obtain the codeword $\mathbf{y}$.

$$\mathbf{x} = (101001), \tag{9}$$

$$\mathbf{e} = (100110\ 000001\ 101010\ 000000), \tag{10}$$

$$\mathbf{y} = \mathbf{x}G + \mathbf{e} = (\overbrace{001111}^{\mathbf{y}_0}\ \overbrace{100101}^{\mathbf{y}_1}\ \overbrace{101101}^{\mathbf{y}_2}\ \overbrace{100100}^{\mathbf{y}_3}). \tag{11}$$

We now decode $\mathbf{y}$ using Alg. 1.

Step 0: Here we simply calculate all possible decodings of the first four bits of $\mathbf{y}_0$, by adding to it all possible error vectors in four bits. This yields the following T_0:

$$T_0 = \{(0011), (0111), (1011), (0010), (0110), (1010), (0001), (0101), (1001)\}.$$

Step 1 - Step 3: Next, we try to remove all vectors in T_0 that does not satisfy (6).

To illustrate the optimization proposed in Remark 1, we look at the processing of the element $\mathbf{x}' = (0011)$ from T_0 in detail through Step 1. As mentioned in the remark, initially we only consider the first two columns of B_1 when trying to determine the validity of $\mathbf{x}'$. That is, we test the following restricted variant of (6):

$$(\mathbf{x}'B_1)[1 \ldots 2] + \mathbf{e}'[1 \ldots 2] \stackrel{?}{=} \mathbf{y}_1[1 \ldots 2], \quad \text{for some } \mathbf{e}' \in E^{k_1/\ell}.$$

With the concrete values of $\mathbf{x}'$, $\mathbf{y}_1$ and B_1 as given above, this becomes:

$$(0011)\begin{pmatrix} 0\,1 \\ 0\,0 \\ 1\,1 \\ 0\,0 \end{pmatrix} + \mathbf{e}'[1 \ldots 2] = (11) + \mathbf{e}'[1 \ldots 2] = (10).$$

By inspection we see that an error of the form $(01XXXX)$ will satisfy this equation, so we continue with the next two columns of B_1:

$$(0011)\begin{pmatrix} 0\,1\,0\,0 \\ 0\,0\,1\,1 \\ 1\,1\,0\,1 \\ 0\,0\,1\,1 \end{pmatrix} + \mathbf{e}'[1 \ldots 4] = (1110) + \mathbf{e}'[1 \ldots 4] = (1001).$$

At this point we see that there are no errors that satisfies this equation (since it would have to be of the form $(01\mathbf{11}XX)$ which is not a valid error in our error set). Hence, we can discard $\mathbf{x}'$ immediately, without considering all of B_1. Applying the same technique to the rest of the elements of T_0 we obtain:

$$L_1 = \{(0010), (1010), (0101)\}.$$

From the elements in L_1 we build up the temporary list T_1 of all possible decodings of $\mathbf{y}_0$ having length $4+1=5$:

$$\begin{aligned} T_1 &= \{\mathbf{x}' \parallel (\mathbf{y}_0[5] + \mathbf{e}') \mid \mathbf{x}' \in L_1,\ \mathbf{e}' \in E^{k_2/\ell}\} \\ &= \{(00100), (00101), (10100), (10101), (01010), (01011)\}. \end{aligned}$$

Repeating the above process for Step 2 and Step 3, we obtain the lists L_2 and L_3:

$$\begin{aligned} L_2 &= \{(10100), (01010)\} \\ L_3 &= \{(101001)\}. \end{aligned}$$

Thus, in this case we obtain a unique decoding.

The efficiency of the list decoding algorithm depends on the size of the lists $L_0, L_1, \ldots, L_w$, and whether during the decoding process each new list has a smaller size than the previous one. If the size of the lists decreases, the overall complexity is dominated by the size of the initial list L_0. Therefore, given a parameter k_1 (which determines L_0), we want to impose constrains on the values of n_i/k_i in order to avoid "blow-up" of the list sizes.

Proposition 3. *Let* $\mathrm{E}[|L_i|]$ *denote the expected value of the size of the lists* $L_1, L_2, \ldots, L_w$. *Then* $|L_0| \geq \mathrm{E}[|L_1|] \geq \cdots \geq \mathrm{E}[|L_w|]$ *if and only if* $\dfrac{n_i}{k_i} \geq \dfrac{\log_2 \rho}{1 - \log_2 \rho}$ *for all* $2 \leq i \leq w$.

Proof. Let $i \in \{1, \ldots, w\}$. Then, from Proposition 2 the condition $\mathrm{E}[|L_{i-1}|] \geq \mathrm{E}[|L_i|]$ turns into

$$\begin{aligned} &\rho^{n_1+k_1+\cdots+n_{i-1}+k_{i-1}} 2^{(k_1+\cdots+k_{i-1})-(n_1+k_1+\ldots n_{i-1}+k_{i-1})} \geq \\ &\geq \rho^{n_1+k_1+\cdots+n_{i-1}+k_{i-1}+n_i+k_i} 2^{(k_1+\cdots+k_{i-1}+k_i)-(n_1+k_1+\ldots n_{i-1}+k_{i-1}+n_i+k_i)} \end{aligned}$$

which in turn is equivalent to

$$2^{n_i} \geq \rho^{n_i+k_i}$$

and further, equivalent to

$$n_i \geq k_i \frac{\log_2 \rho}{1 - \log_2 \rho}.$$

□

4 Application to Encryption and Signatures

In this section we describe how we can construct an encryption and a signature scheme based on the ideas presented in Sect. 2 and Sect. 3. Both schemes share a common description of their key generation, given in Alg. 2.

4.1 Encryption Scheme

The encryption scheme is structurally identical to McEliece, in the sense that for a message $\mathbf{m} \in \mathbb{F}_2^k$, the ciphertext is computed as $\mathbf{c} = \mathbf{m}G_{\mathrm{pub}} + \mathbf{e} \in \mathbb{F}_2^n$. The difference is in the construction of G_{pub} (as defined in Alg. 2), and in the choice of the error vector $\mathbf{e}$ (drawn from a specific set of errors E^m).

Similarly, decryption works by first applying the inverse permutation P^{-1} to the ciphertext, decode the result using Alg. 1, and finally apply the inverse transformation S^{-1}.

4.2 Signature Scheme

Our signature scheme can use the decryption routine directly to sign messages. As mentioned in Ex. 2, for signing purposes, we want the code rate to be high enough so that L_w is likely to be non-empty, whereas for an encryption scheme one generally wants a smaller code rate to obtain unique decoding.

Algorithm 2 Key Generation

Parameters: Let ℓ divide n, $m = n/\ell$ and $E \subset \mathbb{F}_2^\ell$ be an error set of granulation ℓ and density ρ.

Key generation: The following matrices make up the private key:

- An invertible matrix $S \in \mathbb{F}_2^{k\times k}$.
- A permutation matrix $P \in \mathbb{F}_2^{n\times n}$ created as follows. Select a permutation π on $\{1, 2, \ldots, m\}$, and let P be the permutation matrix induced by π, so that for any $\mathbf{y} = \mathbf{y}_1 \parallel \mathbf{y}_2 \parallel \ldots \parallel \mathbf{y}_m \in (\mathbb{F}_2^\ell)^m$:

$$\mathbf{y}P = \mathbf{y}_{\pi(1)} \parallel \mathbf{y}_{\pi(2)} \parallel \ldots \parallel \mathbf{y}_{\pi(m)}, \tag{12}$$

  that is, P only permutes the m substrings of $\mathbf{y}$ of length ℓ.
- A generator matrix G for a binary (n, k) code of the form (5).

Public key: $G_{\text{pub}} = SGP$.

Private key: S, G and P.

Algorithm 3 Decoding for signatures

Input: A vector $\mathbf{y} \in \mathbb{F}_2^n$, and a generator matrix G of the form (5).

Output: A valid decoding $\mathbf{s} \in \mathbb{F}_2^k$ of $\mathbf{y}$.

Procedure:
The notation is the same as in Alg. 1, with the addition of the variables $\mathsf{ExpLimit}_i \leq \rho^{n_i}$. The decoding proceeds in two phases:

Phase 1: Find a valid decoding $\mathbf{x}'$ of $\mathbf{y}_0[1]$ with respect to B_1 and $\mathbf{y}_1$. That is, find an $\mathbf{x}' \in \mathbb{F}_2^{k_1}$ so that $\mathbf{x}'B_1 + \mathbf{y}_1 \in E^{n_1/\ell}$, trying at most $\mathsf{ExpLimit}_1$ candidates. Expand $\mathbf{x}'$ into at most $\mathsf{ExpLimit}_2$ candidates of length $k_1 + k_2$ by appending the sum of $\mathbf{y}_0[2]$ with random errors from $E^{k_2/\ell}$, until you find a valid decoding with respect to B_2 and $\mathbf{y}_2$ (if no valid candidate can be found, start over with a new initial $\mathbf{x}'$). Continue this process for $(B_3, \mathbf{y}_3)$, $(B_4, \mathbf{y}_4), \ldots$

Phase 2: Once you have found a candidate that is valid for $B_1, B_2, \ldots, B_{w-1}$ and $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{w-1}$, switch to the list-decoding algorithm described in Alg. 1 for the last block, i.e. for $i = w - 1$.

Return: $\mathbf{s} \leftarrow L_w$.

However, for signing, we actually don't need to find all the possible decodings for a certain syndrome $\mathbf{y} \in \mathbb{F}_2^n$, as described in Alg. 1; we need only one. That is why we need an alternative decoding algorithm that, with very high probability, will find only one decoding.

We now present a randomized decoding algorithm, Alg. 3, that is suitable for finding digital signatures. In Sect. 5.3 we will see further security related justification for this alternative algorithm.

Since Alg. 3 is a randomized version of the Alg. 1 we need a condition that guarantees that the signing process will find a signature with high probability.

Proposition 4. *Algorithm 3 produces a signature with probability more than* $1/2$ *if the following two conditions hold:*

1. $\mathsf{ExpLimit}_i > (2/\rho)^{n_i}$, *for* $1 \leq i \leq w - 1$;
2. $\rho^{k_w+n_w}2^{-n_w} > 1$.

Proof. 1. In each step of the Alg. 3, Phase 1, we need to find a valid extension of the error by picking a valid part of length k_i and then validating it on the remaining n_i bits. The probability that the n_i bits, when the corresponding part of x is evaluated, to give a valid error extension is

$$(\frac{|E|}{2^l})^{n_i/\ell} = (\frac{\rho}{2})^{n_i}$$

Hence, we need approximately $\mathsf{ExpLimit}_i \approx (2/\rho)^{n_i}$ tries in order to find a valid error extension.

2. Since the List decoding for producing signatures starts in the last block, the claim follows directly from Proposition 2.

□

For concreteness, we describe the signing and verification procedures in detail in Alg. 4 and 5, respectively.

Algorithm 4 Signing

Input: A value $\mathbf{z} \in \mathbb{F}_2^n$ to be signed. The private key S, G and P.

Output: A valid signature $\boldsymbol{\sigma} \in \mathbb{F}_2^k$, so that $\boldsymbol{\sigma} G_{\text{pub}} + \mathbf{z} \in E^m \subset \mathbb{F}_2^n$.

Procedure:

1. Compute $\mathbf{y} = \mathbf{z}P^{-1}$.
2. Decode $\mathbf{y}$ using Alg. 3, to get a valid decoding $\mathbf{s}$.
3. Set the signature $\boldsymbol{\sigma} = \mathbf{s}S^{-1}$.

Algorithm 5 Verification

Input: A pair $(\mathbf{z}, \boldsymbol{\sigma}) \in \mathbb{F}_2^n \times \mathbb{F}_2^k$, and the public key G_{pub}.

Output:

$$\text{Ver}(\mathbf{z}, \boldsymbol{\sigma}) = \begin{cases} \text{Accept}, & \text{if } \boldsymbol{\sigma} G_{\text{pub}} + \mathbf{z} \in E^m \subset \mathbb{F}_2^n. \\ \text{Reject}, & \text{otherwise.} \end{cases}$$

5 Security Analysis

The security of code-based systems relies on the hardness of finding a codeword that is closest to a certain word, given that such a codeword exists. There are two approaches for solving this problem: generic decoding algorithms that assume no knowledge about the structure of the code, and structural attacks that try to exploit the known structure of the code.

5.1 Information Set Decoding for Error Sets of a Given Density

The best generic decoding algorithms, when assuming a random code of a given rate and known error set, are based on Information-Set Decoding (ISD). The technique was introduced by Prange [41], and later improved several times in the works of Lee and Brickell [30], Leon [31], Stern [44], and many others [20,12,35,7].

In essence, the idea behind all ISD algorithms is the following. Find an information set $\mathcal{I}$ i.e. an index set of k columns of the generator matrix G that form an invertible matrix $G_{\mathcal{I}}$, such that the error vector has a specific error pattern $\mathbf{e}_{\mathcal{I}}$ with respect to $\mathcal{I}$. With the error pattern being correctly guessed, we can find the message as $\mathbf{m} = (\mathbf{c}_{\mathcal{I}} + \mathbf{e}_{\mathcal{I}})G_{\mathcal{I}}^{-1}$, where $\mathbf{c}_{\mathcal{I}}$ is the part of the ciphertext $\mathbf{c}$ corresponding to the information set $\mathcal{I}$. In our case, the error vector is not characterized by its Hamming weight, but by the density of the error set out of which it was drawn. Nevertheless, the idea and techniques of ISD can still be successfully applied with an appropriate adaptation. We call this the *ρISD problem*.

Let an error set E_ℓ of density ρ and granulation ℓ, with $m\ell = n$, be used in the coding process, i.e. the error vector is randomly picked from the set E_ℓ^m. Note that WLOG we can assume that the "all-zero" error $\mathbf{0}$ is in E_ℓ. Also, for simplicity, we assume that ℓ divides k. In our adaption of the plain ISD attack we hope that there are no 1's in the part of the error vector corresponding to our chosen information set. In other words, we hope that $\mathbf{0} \in E_\ell$ has been used for all of the k/ℓ blocks corresponding to $\mathcal{I}$. (More generally, this can be seen as guessing exactly the k/ℓ errors in $\mathcal{I}$, no matter what they actually look like with respect to the Hamming metric.) The probability of success is $\frac{1}{|E|^{k/\ell}} = \rho^{-k}$.

Similarly, in our analogue of the Lee-Brickell variant, we allow p blocks to have a different error pattern than $\mathbf{0}$, so the probability of success is $\binom{k/\ell}{p}\left(\frac{|E|-1}{|E|}\right)^p\left(\frac{1}{|E|}\right)^{k/\ell-p} = \binom{k/\ell}{p}\frac{(\rho^\ell-1)^p}{\rho^k}$.

Similar adaptions to our setting can be made for all the various variants of ISD as follows.

Let ρISD_{VAR} denote the complexity of some variant of the ISD algorithms adapted to error sets of density ρ. Then as usual, we can write:

$$\rho ISD_{VAR} = \rho Pr_{VAR}^{-1} \cdot \rho Cost_{VAR}$$

where ρPr_{VAR} is the probability of success of one iteration of the algorithm, and $\rho Cost_{VAR}$ denotes the cost of each of the iterations. We summarize the results of adapting several ISD variants in the following theorem.

Theorem 2. *The probability of success of one iteration and the cost of one iteration of the Lee-Brickell variant, Stern variant, Finiasz-Sendrier variant, Bernstein-Lange-Peters variant, May-Meurer-Thomae variant and Becker-Joux-May-Meurer variant adapted to error sets of density ρ are given in Table 1.*

Proof (sketch). We first note that all the parameters and the strategy used in the presented variants ρISD_{VAR} is the same as in the original algorithms ISD_{VAR}. The main difference is in the probability of success and the size of the constructed lists.

In the original variants, one allows a certain amount of errors to appear in the coordinates indexed by the information set $\mathcal{I}$ in a specific pattern. Also, a specific number of errors is allowed on certain coordinates outside $\mathcal{I}$. Each and every new algorithm uses different pattern, carefully chosen in order to increase the probability of having such a particular pattern compared to previous variants.

Without loss of generality, let $\tilde{k}$ be the size of some fixed portion of the coordinates. From the discussion at the beginning of the section, we see that unlike in the standard ISD_{VAR}, in ρISD_{VAR} the probability of "guessing" the pattern in p blocks does not depend on n, or the structure of the error vector outside the fixed coordinates. It is always given by $\binom{\tilde{k}/\ell}{p}\frac{\left(\rho^\ell-1\right)^p}{\rho^{\tilde{k}}}$. This probability can be used to compute the probability of any of the variants.

Now, assuming that the pattern is correctly guessed, one forms one or more lists of size p subsets of the fixed coordinates, in order to match a computed tag to another, using plain or collision type of matching. In our case, the number of such subsets is given by $\binom{\tilde{k}/\ell}{p}\left(\rho^\ell-1\right)^p$, where $\binom{\tilde{k}/\ell}{p}$ is the number of size p subsets of blocks of length ℓ, and $\left(\rho^\ell-1\right)$ is the number of possible error patterns inside a block where we allow to have not guessed the pattern. Using this formula we can compute the size of the created lists in the algorithms of all variants. The particular details are left to the reader.

Table 1. Complexity of ISD variants adapted to error sets of density ρ. $Cost_{Gauss}$ denotes the complexity of Gaussian elimination. The meaning of the optimizing parameters in each of the formulas below can be found in [30,44,20,12,35,7].

Variant	ρPr_{VAR}	$\rho Cost_{VAR} - Cost_{Gauss}$
LB	$\binom{k/\ell}{p}\frac{(\rho^\ell-1)^p}{\rho^k}$	$\binom{k/\ell}{p}(\rho^\ell-1)^p pn$
ST	$\binom{k/2\ell}{p}^2\frac{(\rho^\ell-1)^{2p}}{\rho^{k+\lambda}}$	$2\lambda pL + 2pn\frac{L^2}{2^\lambda}, \quad L = \binom{k/2\ell}{p}(\rho^\ell-1)^p$
FS	$\binom{(k+\lambda)/2\ell}{p}^2\frac{(\rho^\ell-1)^{2p}}{\rho^{k+\lambda}}$	$2\lambda pL + 2pn\frac{L^2}{2^\lambda}, \quad L = \binom{(k+\lambda)/2\ell}{p}(\rho^\ell-1)^p$
BLP	$\binom{k/2\ell}{p}^2\binom{\lambda_1/\ell}{q}\binom{\lambda_2/\ell}{q}$ $\cdot\frac{(\rho^\ell-1)^{2p+2q}}{\rho^{k+\lambda_1+\lambda_2}}$	$\binom{k/2\ell}{p}(\rho^\ell-1)^p 2(\lambda_1+\lambda_2)p + \binom{k/2\ell}{p}\left(\binom{\lambda_1/\ell}{q}+\binom{\lambda_2/\ell}{q}\right)(\rho^\ell-1)^{p+q}(\lambda_1+\lambda_2)q$ $+\frac{\binom{k/2\ell}{p}^2\binom{\lambda_1/\ell}{q}\binom{\lambda_2/\ell}{q}(\rho^\ell-1)^{2p+2q}}{2^{\lambda_1+\lambda_2}}2(p+q)n$
MMT	$\binom{(k+\lambda)/2\ell}{p}^2\frac{(\rho^\ell-1)^{2p}}{\rho^{k+\lambda}}$	$2\lambda_2 pL + (2n+\lambda-\lambda_2)p\frac{L^2}{2^{\lambda_2}} + pn\frac{L^4}{2^{\lambda+\lambda_2}}, \quad L = \binom{(k+\lambda)/2\ell}{p/2}(\rho^\ell-1)^{p/2}$
$BJMM$	$\binom{(k+\lambda)/\ell}{p}\frac{(\rho^\ell-1)^p}{\rho^{k+\lambda}}$	$4Pr_{coll}^{-4}p_2\left(L_3\log_2 R_2 + n\frac{L_3^2}{R_2}\right) + 2n\left(p_1\frac{L_2^2R_2}{R_1} + p\frac{L_1^2R_1}{2^\lambda}\right),$ $Pr_{coll} = \binom{(k+\lambda)/2\ell}{p_2/2}^2\binom{(k+\lambda)/\ell}{p_2}^{-1},\ p_i = \frac{p_{i-1}}{2}+\epsilon_i,\ i=1,2, p_0=p,$ $L_i = \binom{(k+\lambda)/2\ell}{p_i}(\rho^\ell-1)^{p_i}, i=1,2, \quad L_3 = \binom{(k+\lambda)/2\ell}{p_2/2}(\rho^\ell-1)^{p_2/2},$ $R_i = \binom{p_{i-1}}{p_{i-1}/2}\binom{(k+\lambda)/\ell-p_{i-1}}{\epsilon_i}(\rho^\ell-1)^{\epsilon_i},\ i=1,2, p_0=p$

□

In Table 2 we state concrete complexities of the various adaptions for the concrete parameters $\ell = 2$ and $\rho = 3^{1/2}$, when $k = 256$ or $k = 512$.

5.2 Modelling ρISD using Polynomial System Solving

In this part we describe how the ρISD problem can be modeled as the Polynomial System Solving (PoSSo) problem. PoSSo is the problem of finding a solution to a system of polynomial equations over some field.

Table 2. Complexity of ISD variants for $\ell = 2$, $\rho = 3^{1/2}$ when $k = 256$ and $k = 512$.

Variant	LB	ST	FS	BLP	MMT	$BJMM$
$k = 256$	2^{212}	2^{197}	2^{186}	2^{186}	2^{146}	2^{123}
$k = 512$	2^{416}	2^{381}	2^{356}	2^{356}	2^{279}	2^{226}

Given a public generator matrix G_{pub} and a ciphertext $\mathbf{c}$, we can form n linear equations

$$G_{\text{pub}} + \mathbf{y} = \mathbf{c},$$

where $\mathbf{x}$ denotes the k unknown bits of the message, and $\mathbf{y}$ is the n-bit unknown error. Clearly, we don't have enough equations to find the correct solution efficiently. However, from the known structure of the error vector we can derive additional equations of higher degree that describe exactly the error set. If we denote these equations as $P(\mathbf{y}) = 0$, then a solution of the system

$$\begin{aligned} \mathbf{x}G_{\text{pub}} + \mathbf{y} &= \mathbf{c} \\ P(\mathbf{y}) &= 0 \end{aligned} \tag{13}$$

will give the same solution for the message and the error vector as the decoding algorithm with the knowledge of the private key.

We emphasize that any error set can be described by a system of equations, including the set of errors of a bounded weight used in the McEliece system. The efficiency of this approach strongly depends on the error structure.

Remark 2. Note that, in order for a choice of error set to be secure, the set of polynomials S used to define it should not contain any linear polynomials (nor be isomorphic to a such a set). Without this restriction, the system in (13) becomes easily solvable.

Furthermore, it is possible to introduce an optimization parameter in the form of a guess of some of the errors, or a guess of linear equations for the errors. In what follows we present the modeling of an error set of density $\rho = 3^{1/2}$ and granulation $\ell = 2$.

Let E_ℓ be an error set of density $\rho = 3^{1/2}$ and granulation $\ell = 2$. Without loss of generality, we can assume that $E_\ell = \{(00), (01), (10)\}$. Let $(e_1, e_2) \in E_\ell$. Then, the equation $e_1 e_2 = 0$ describes completely the error set E_ℓ. Hence, the system (13) turns into:

$$\begin{aligned} (x_1, \ldots, x_k)G_{\text{pub}} + (y_1, \ldots, y_n) &= \mathbf{c} \\ y_1 y_2 &= 0 \\ &\ldots \\ y_{n-1} y_n &= 0 \end{aligned}$$

The system can be easily transformed to the following form:

$$\begin{aligned} A_1(x_1, \ldots, x_k)A_2(x_1, \ldots, x_k) &= 0 \\ &\ldots \\ A_{n-1}(x_1, \ldots, x_k)A_n(x_1, \ldots, x_k) &= 0 \end{aligned} \tag{14}$$

where A_i are some affine expressions in the variables $x_1, \ldots, x_k$.

We can introduce an optimization parameter p as follows. Suppose we have made a correct guess that the equation $y_{2t-1} + y_{2t} = b_t$, $b_t \in \{0, 1\}$ holds for p pairs (y_{2t-1}, y_{2t}) of coordinates of the error vector. Adding these p new equations to the system reduces the complexity of solving it. Note that it is enough to correctly guess k equation to obtain a full system of k unknowns. The probability of making the correct guess is $Pr = (2/3)^p$. Under the natural constrain $0 \leq p \leq k$, we can roughly estimate the complexity to

$$Comp = (2/3)^p \cdot \left(\binom{k-p}{Dreg_{k-p}} + p \right)^\omega$$

where $Dreg_{k-p}$ denotes the degree of regularity of a system of $k-p$ variables of the form (14).

We performed some experiments using the F_4 algorithm [19] implemented in MAGMA [34], and based on rather conservative projections of the degree of regularity, we give the following table with a rough estimate of the lower bound of the complexity.

Table 3. Estimated complexity of solving ρISD using the F_4 algorithm for $\ell = 2$, $\rho = 3^{1/2}$.

k	Complexity
128	2^{84}
256	2^{152}
512	2^{237}

5.3 Distinguishing Attacks

A distinguishing attack on our scheme, will in essence try to recover a decomposition of the public key G_{pub} into $G_{\text{pub}} = S'G'P'$, where G' has a shape similar to that of (5). We emphasize that the attacker does not have to recover G exactly, but rather a similar G'. Once G' is obtained, the attacker can try to perform decoding as the normal user. The complexity of the message recovery will depend on the level of structural similarity of obtained equivalent key G' with the original key G.

Let $[\frac{n}{\ell}]$ denote the set $\{1, 2, \ldots, \frac{n}{\ell}\}$, and for $\mathcal{I} \subset [\frac{n}{\ell}]$ denote by $(G_{\text{pub}})_{\mathcal{I}}$ the $|\mathcal{I}|$ corresponding blocks of ℓ columns picked from G_{pub}. Further, let $K_i = k_1 + \cdots + k_i$ and $N_i = K_i + n_1 + \ldots + n_i$. We partition the public key as $G_{\text{pub}} = (G'_X \; G'_Y)$, where the submatrices G'_X and G'_Y have dimension $k \times n_X$ and $k \times n_Y$ respectively and initially $n_X = 0$ and $n_Y = n$.

Step 1. The crucial idea is to notice that there exist submatrices of blocks of columns that have smaller rank than expected. Let $\mathcal{I}_1$ denote the coordinate set of $\lceil \frac{K_t}{\ell} + 1 \rceil$ randomly selected blocks of columns from G_{pub}, where $1 \leq t < w$ is an optimization parameter. This parameter t allows us to find the best choice of the set $\mathcal{I}_1$ that gives the lowest complexity of the attack. For a random matrix we would expect, with high probability, that $\text{rank}(G_{\text{pub}})_{\mathcal{I}_1} = K_t + \ell$. However, due to the structure of G, we hope to find columns such that:

$$\text{rank}(G_{\text{pub}})_{\mathcal{I}_1} < K_t + \ell. \tag{15}$$

After finding such an $\mathcal{I}_1$, we set $G'_X \leftarrow (G_{\text{pub}})_{\mathcal{I}_1}$ and $G'_Y \leftarrow (G_{\text{pub}})_{[\frac{n}{\ell}] \setminus \mathcal{I}_1}$.

Step 2. After Step 1 is finished, we perform a greedy selection among the remaining column blocks, incrementally expanding G'_X (and simultaneously shrinking G'_Y) with the blocks that give the minimum rank (of G'_X), until $\text{rank}(G'_X) = k$. That is, we pick single block column sets $\mathcal{I}_2 \subset [\frac{n}{\ell}] \setminus \mathcal{I}_1$, $\mathcal{I}_3 \subset [\frac{n}{\ell}] \setminus \mathcal{I}_1 \cup \mathcal{I}_2$, $\ldots$, so that the ranks of $G'_X \leftarrow (G_{\text{pub}})_{\mathcal{I}_1,\mathcal{I}_2}$, $G'_X \leftarrow (G_{\text{pub}})_{\mathcal{I}_1,\mathcal{I}_2,\mathcal{I}_3}$, $\ldots$, are minimal.

Step 3. Since $1 \leq t < w$, the steps so far will distinguish the union of the columns of the matrices $B_{t+1}, \ldots, B_w$ and the $k_{t+1} + \cdots + k_w$ columns from I_k. This is not enough to distinguish the code, as t can be close to w. Hence we need to repeat the Steps 1 and 2 for the smaller code composed of the remaining non-distinguished columns.

Step 4. In the last step, we recover the matrices P' and S'. Let P'^{-1} denote the permutation matrix corresponding to the columns selected during Steps 2, i.e.:

$$G' = G_{\text{pub}}P'^{-1} = (G'_X \; G'_Y) = \left((G_{\text{pub}})_{\mathcal{I}_1,\ldots,\mathcal{I}_{n/\ell}} \; (G_{\text{pub}})_{[\frac{n}{\ell}] \setminus \bigcup_i \mathcal{I}_i} \right). \tag{16}$$

The G' in (16) has an "internal" block structure, of $\frac{n}{\ell}$ blocks. Still, it is not of the form as the matrix in (5), since it it does not have the step-wise structure with zeroes below each block. However, by performing some elementary row operations on G' we can get it into the right form which allows decoding.

Note that the G' found by the above process is not in systematic form as is the matrix in (5). It can be brought to form (5) using a standard procedure for obtaining the systematic form, having in mind that only block column operations are allowed.

Remark 3. A distinguishing attack can also be performed on the generator matrix of the dual code, since it also has a stepwise structure. The procedure described above will be the same, but the roles of the parameters k_i and n_i will be interchanged as follows $k_1 \mapsto n_w, \ldots, k_w \mapsto n_1$, $n_1 \mapsto k_w, \ldots, n_w \mapsto k_1$.

The following theorem estimates the complexity of the distinguishing attack.

Theorem 3. *The complexity of the distinguishing attack is the minimum of the attacks on the code and the dual code, i.e.,*

$$Dist = min\{Pr_{rank}^{-1} \cdot Cost, \quad Pr_{rankD}^{-1} \cdot Cost_D\} \tag{17}$$

where $Cost = k(K_t + \ell)^{\omega-1}$, $Cost_D = (n-k)(N_t + \ell)^{\omega-1}$, ω *is the linear algebra constant, and*

$$Pr_{rank} = \binom{n/\ell - (K_t/\ell + 1)}{N_t/\ell - (K_t/\ell + 1)} \binom{n/\ell}{N_t/\ell}^{-1}, \quad Pr_{rankD} = \binom{n/\ell - ((N_t - K_t)/\ell + 1)}{N_t/\ell - ((N_t - K_t)/\ell + 1)} \binom{n/\ell}{N_t/\ell}^{-1}. \tag{18}$$

Proof. We will evaluate the complexity of the attack on the generator matrix of the code. The attack on the dual code is analogous.

First of all, lets emphasize that Step 1 has the biggest complexity, so we can WLOG omit the other steps from our complexity estimation.

First, let's consider a more general strategy of choosing in the attack. Suppose we first choose m/ℓ blocks, $K_t < m < n$ out of the possible n/ℓ in the hope that, among them there are $K_t/\ell + 1$ blocks with smaller rank than $K_t + \ell$. The probability of this to happen is

$$Pr_m = \binom{m/\ell}{K_t/\ell + 1} \binom{n/\ell - m/\ell}{N_t/\ell - (K_t/\ell + 1)} \binom{n/\ell}{N_t/\ell}^{-1} \tag{19}$$

Suppose the choice was made correctly. Now the cost of finding the $K_t/\ell + 1$ blocks of smaller rank among the m/ℓ is

$$Cost_m = \binom{m/\ell}{K_t/\ell + 1} \cdot Cost_{rank}$$

Hence, the total amount of work is

$$Dist_m = Pr_m^{-1} \cdot Cost_m = \binom{n/\ell}{N_t/\ell} \binom{n/\ell - m/\ell}{N_t/\ell - (K_t/\ell + 1)}^{-1} \cdot Cost_{rank}$$

Since $m/\ell \geq (K_t/\ell + 1)$, the minimum complexity is obtained for $m = K_t/\ell + 1$. From here we get that

$$Pr_{K_t/\ell+1} = Pr_{rank} = \binom{n/\ell - (K_t/\ell + 1)}{N_t/\ell - (K_t/\ell + 1)} \binom{n/\ell}{N_t/\ell}^{-1} \tag{20}$$

and

$$Cost_{K_t/\ell+1} = Cost_{rank}$$

The rank computation takes approximately $Cost_{rank} = k(K_t + \ell)^{\omega-1}$ operations, where ω is the linear algebra constant.

As we said, the same strategy applies for the dual code, for which instead of K_t we have $N_t - K_t$. Now the claim follows directly.

The parameter t is an optimization parameter for the attack, and its optimal value depends strongly on the chosen parameters. We should note that, for the sets of parameters that we use, because of the nature of the curve of the complexity for different K_t, the best strategies are always for either $t = 1$ or $t = w - 1$. □

We have noticed that the best complexity of the attack is achieved when the optimization parameter t is either 1 or $w - 1$.

Remark 4. We emphasize that the generator matrix of *any* linear (n, k) code can be transformed to the form (5) using the attack described above. Since the size of the K_i is not known, one would use a trial-and-error approach, starting from some chosen small K_1, and slowly increasing its value, until a smaller rank than expected is distinguished. The size of the obtained K_1 depends on the code, and for randomly selected code it is expected to depend on the dimension k.

5.4 Cheap Distinguishing Attacks With Expensive Recovery/Forgery

Here we describe two distinguisher attacks, having relatively small complexity. As in the previous analysis, these distinguishers lead to recovery of an alternative private key. However, the complexities of decoding or signature forgery with these recovered private keys are very expensive, i.e. infeasible.

Attack on the encryption scheme: Denote by $\mathcal{C}$ the (n, k) code of the generator matrix G in (5), and by $\mathcal{D}$ the dual (n, r) code of $\mathcal{C}$. Up to a permutation of coordinates, the code $\mathcal{D}$ admits a subcode spanned by the $n_1 \times r$ matrix $A = [B_1^T \mid I_d \mid 0 \dots 0]$, where I_d is the identity matrix, and B_1^T is the transpose of the block B_1 in G. Matrix A has $n - n_1 - k_1$ columns of zeros in its rightmost positions. These positions can be ignored as far as we consider codewords obtained by combining the rows of A. Consequently, the minimum distance of $\mathcal{D}$ is not greater than the minimum distance of the $(n_1 + k_1, n_1)$ code spanned by $[B_1^T \mid I_d]$. Note that a random $(n_1 + k_1, n_1)$ linear binary code has an expected minimum distance d_{min} which equals the Gilbert—Varshamov bound [6]:

$$d_{min} = (n_1 + k_1) \cdot \delta_{GV}(\frac{n_1}{n_1 + k_1}), \tag{21}$$

where $\delta_{GV} \leq \frac{1}{2}$ is the relative Gilbert–Varshamov (GV) distance of a code having rate R. It is defined as being the root of $F(x) = H_2(x) - 1 + R$, where $H_2(x)$ is the binary entropy function. Thus, the dual codes $\mathcal{D}$ of our scheme will have a minimum distance of $d_{min} = (n_1 + k_1) \cdot \delta_{GV}(\frac{n_1}{n_1+k_1})$, instead of $d_{min,rand} = n \cdot \delta_{GV}(\frac{r}{n})$.

In Sect. 6 and Appx. A we consider a wide range of choices for the parameters n, k, n_i and k_i. For the dual codes, we can approximate the cost of finding a word of weight d_{min} for these choices, by applying Stern's algorithm [44]:

$$\text{Cost}_{n,r,d_{min}} \approx 2^{a_0 + a_1 n + a_2 r + a_3 d_{min} + a_4 n d_{min} + a_5 n k + a_6 k d_{min} + a_7 d_{min}^2}, \tag{22}$$

where $a_0 = 20.0482$, $a_1 = 0.00519929$, $a_2 = 0.0019762$, $a_3 = 0.31586$, $a_4 = -0.000211907$, $a_5 = -1.15349 \times 10^{-6}$, $a_6 = 0.000536886$ and $a_7 = 0.00610952$.[6]

For example, let us consider the code defined by $(n, k) = (7590, 1278)$, $w = 155$, $K = (46, 8, 8, \dots, 8)$, $N = (32, 32, \dots, 32, 1384)$, with dual code $(7590, 6312)$. Since $n_1 = 32$ and $k_1 = 46$, a random binary $(n_1 + k_1, n_1) = (78, 32)$ linear code would be expected to have a minimum distance of 12. From the approximation in (22), we have that finding a word of weight 12 in the binary code $(7590, 6312)$ is going to cost around 2^{43} binary operations.

Now, the main question is: *How good is the equivalent key that the attacker can get from this distinguisher?*

We answer this question with the following analysis. Let $\mathbf{w} \in \mathbb{F}_2^n$ be the codeword of weight d_{min} that was obtained with the above distinguishing method against the dual code with parameters (n, r). Note that the support of $\mathbf{w}$ (non-zero coordinates) is dispersed among n columns. In order to build an equivalent key G' with a stepwise form similar to that of G in (5), the attacker needs to process the whole first block B_1 of G by calling the distinguisher n_1 times and finding codewords with weight d_{min}. The total size of the obtained support in this phase determines the value k_1' in the equivalent key G' and is crucial for the complexity of decoding with G'. The maximal value for k_1' is $n_1 d_{min}$, and a lower bound is given in (23) (modeled as the expected number of non-zero coordinates in a vector of n coordinates with an equiprobable dispersion of $n_1 d_{min}$ ones; and reduced by n_1, corresponding to the I_d part in A):

$$k_1' = n - \frac{(n-1)^{n_1 d_{min}}}{n^{n_1 d_{min} - 1}} - n_1. \tag{23}$$

Consequently, the decoding procedure using G' will have a complexity of around $O(3^{\frac{k_1'}{2}})$ operations.

For the $(n, k) = (7590, 1278)$ code considered in the example above, this cheap distinguishing attack will need around $\approx 2^{246}$ operations just for the start of the decoding procedure.

[6] We note that it is possible to get slightly better distinguishing complexities with the BJMM algorithm [7] but it was harder for us to obtain a closed approximation formula as (22) since in [7] the analysis is for another fixed weight value of d_{min}.

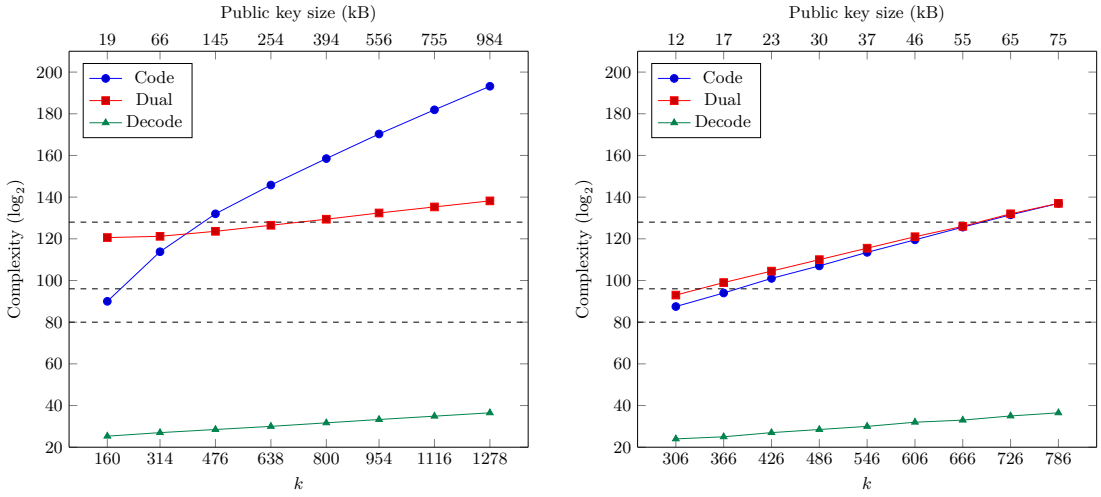
Attack on the signature scheme: Someone can apply the key recovery attack described in [29] for the signature scheme based on convolutional codes described in [33]. In the first look, this makes sense, since someone can see the similarities between our scheme and the scheme in [33]. However, there are two crucial differences that makes this attack infeasible:

1. The structure of the secret key [33] is such that for its decoding, direct decoding techniques for convolutional codes in the Hamming metrics are applied. On the other hand, we do not use the Hamming metrics in our codes, but a metrics that involves granularity with a density $\rho = 3^{1/2}$. Consequently, the recovered equivalent key by the attack described in [29] will not have the property described in Proposition 3 : $\frac{n_i}{k_i} \geq \frac{\log_2 \rho}{1 - \log_2 \rho}$ for all $2 \leq i \leq w$. That will make the recovered equivalent key useless for performing feasible list-decoding.
2. The authors of the attack paper [29] suggest at the end of their work that the scheme in [33] can become resistant to their key recovery attack if they add additional random code et the end of their code in lengths of around 140 bits. Interestingly, our codes for signatures have already that random part at the end of the code.

Finally, we emphasize that in many public-key schemes, distinguishing a public-key from random is not considered an attack nor a flaw. This is the case with our scheme too. It would become an attack if the distinguisher leads to the recovery of a private key that facilitates efficient decryption.

6 Choosing Parameters

One important issue with any cryptographic primitive is its efficiency for a given level of claimed security. For public-key primitives, this can be examined by analyzing the sizes of the private and public key, and the number of operations necessary for encryption, decryption, signing and verification.



(a) Encryption (b) Signature

Fig. 5. Comparison between the complexity of decoding and the distinguishing attacks for encryption and signature. Dashed horizontal lines denote three security levels: 2^{80}, 2^{96} and 2^{128}.

From the analysis in Sect. 5.3 we have chosen a set of eight parameters for encryption and nine parameters for signatures, with security levels in the range of $2^{80} - 2^{128}$ (actually slightly above 2^{128}) according to Thm. 3. The full description of the proposed parameters are in Appx. A. In Fig. 5 we plot comparative curves for the complexities of decoding and the complexities of the distinguishing attacks on the code and its dual.

Fig. 6 illustrates how the complexities in Fig. 5a were calculated, by considering the fifth data point in the graph, representing a $(800, 4840)$ code suitable for encryption. By examining all possible values of the attack parameter K_t we find that the distinguishing attack on the code and its dual (a code of dimensions $(4040, 4840)$) has a minimum complexity of 2^{129}.

Similarly, in Fig. 7, we illustrate with a code suitable for signatures (it is the $(786, 1578)$ code in point eight of Fig. 5b). Again, by examining all possible values of the attack parameter K_t we see that the minimum complexity of the attack (on either the code or its dual $(792, 1578)$ code) is 2^{137}.

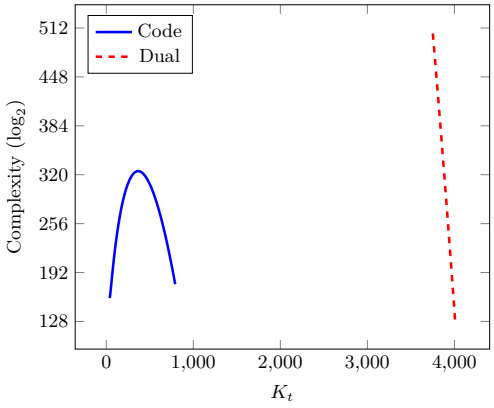


Fig. 6. Complexity of the distinguishing attack from Sect. 5.3 on a $(800, 4840)$ code suitable for encryption (and the dual $(4040, 4840)$ code), for various choices of K_t.

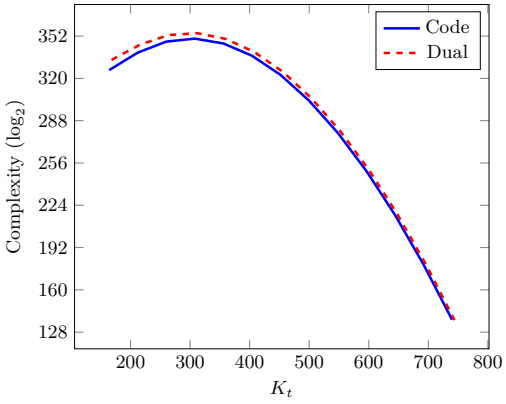


Fig. 7. Complexity of the distinguishing attack from Sect. 5.3 on a $(786, 1578)$ code suitable for signatures (and the dual $(792, 1578)$ code), for various choices of K_t.

Note that the gap between the attack complexity and the complexity of decoding in Fig. 5a (the encryption scheme) is almost a constant, and in Fig. 5b (the signature scheme) the gap is just slowly increasing. While this can be arguably considered as a negative characteristic of the scheme, we want to emphasize the following two arguments why we see our scheme as potentially a useful cryptographic design: 1. The attack complexities are in the stratosphere of infeasibility of "real world" computations in the levels of $2^{80} - 2^{128}$ (or slightly above), while the decryption complexities are in the feasible levels of $2^{23} - 2^{36}$; 2. Since our decoding procedure has the feature of being trivially parallelizable, it is feasible to reduce the decoding complexity from 2^{36} time units down to only a few time units. If the same amount of parallel computing power is given to the attacker, the reduction in the attack complexities will be much smaller[7], thus keeping the complexities of attacks utilizing parallelism in the stratosphere of infeasibility .

7 Conclusions

We have introduced a cryptographic communication channel where the sender has the role of the "noise" and can produce error vectors from an almost arbitrary big error set. For those error sets we defined a new family of binary linear codes that with overwhelming probability can be decoded by an efficient list decoding algorithm.

Having introduced the new codes, we constructed both encryption and signature schemes that follow the basic structure of the McEliece scheme: $G_{\text{pub}} = SGP$. We showed that the security of our schemes are tightly connected to the problem of decoding a random syndrome in the Hamming metric by providing an analog to the Information Set Decoding techniques for our error sets. Further, we scrutinized the power of rank attacks against our scheme and that resulted to a particular choice of parameters that offer a security in the range $2^{80} - 2^{128}$ with plausible operating characteristics. Finally, we analyzed distinguishing attacks that can recover some equivalent key, and showed that those recovered keys are useless for the message recovery or signature forgery since they do not reassemble the subtle structure of our codes (a requirement that guarantees that the list-decoding procedure is stable, convergent and feasible).

We point out to some research directions and open questions connected with our schemes: 1. Finding parameter sets that will offer security levels in the range of 2^{256}, 2. Reducing the public key sizes with techniques such as cyclic and MDPC codes. 3. Implementations in hardware making heavy use of the inherent parallelism in the decoding algorithm for our codes.

[7] Decoding procedures use much simpler matrix-vector multiplications, while the rank attacks have to perform infeasible number of more expensive operations of matrix rank computations.

Acknowledgements

We would like to thank the Lorentz Center at the University of Leiden and Tanja Lange for their invitation to Danilo Gligoroski to attend the workshop "Post-Quantum Cryptography and Quantum Algorithms". There he had first useful discussion with Christiane Peters (to whom we also owe a big acknowledgement). A big acknowledgement also goes to Scholar Visitor Program of the NordSecMob - Erasmus Mundus Master Program, under which Sergey Bezzateev visited NTNU and where this paper was first initiated. We would like to thank various anonymous reviewers for EUROCRYPT 2014, CRYPTO 2014 and PQCrypto 2014 that pointed out to us the cheap distinguishing attacks.

References

1. Mohssen Alabbadi and Stephen B. Wicker. Cryptanalysis of the Harn and Wang modification of the Xinmei digital signature scheme. In *Electronics Letters 28,*, pages 1756–1758, 1992. (Cited on page 3.)
2. Mohssen Alabbadi and Stephen B. Wicker. Security of Xinmei digital signature scheme. In *Electronics Letters 28,*, pages 890–891, 1992. (Cited on page 3.)
3. Mohssen Alabbadi and Stephen B. Wicker. A digital signature scheme based on linear error-correcting block codes. In *Josef Pieprzyk and Reihanah Safavi-Naini (editors). Advances cryptology-ASIACRYPT '94. Proceedings of the Fourth International Conference held at the University of Wollongong, Wollongong, November 28-December 1, Lecture Notes Computer Science 917. Springer*, pages 238–248, 1994. (Cited on page 3.)
4. D. Angluin and L.G. Valiant. Fast probabilistic algorithms for Hamiltonian circuits and matchings. *J. of Computer and System Sciences*, 19:155–193, 1979. (Cited on page 6.)
5. M. Baldi, F. Chiaraluce, R. Garello, and F. Mininni. Quasi-cyclic Low-Density Parity-Check Codes in the McEliece cryptosystem. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 951–956, 2007. (Cited on page 3.)
6. Alexander Barg and G. David Forney Jr. Random codes: Minimum distances and error exponents. *IEEE Transactions on Information Theory*, 48(9):2568–2573, 2002. (Cited on page 17.)
7. 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 *Proceedings of the 31st Annual international conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'12, pages 520–536, Berlin, Heidelberg, 2012. Springer-Verlag. (Cited on pages 3, 12, 13, and 17.)
8. Elwyn Berlekamp, Robert J. McEliece, and Henk C. A. Van Tilborg. On the inherent intractability of certain coding problems. *Information Theory, IEEE Transactions on*, 24(3):384–386, 1978. (Cited on page 7.)
9. D. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. *Post-Quantum Cryptography*, pages 31–46, 2008. (Cited on page 3.)
10. Daniel J. Bernstein. List decoding for binary goppa codes. In *Coding and cryptology—third international workshop, IWCC 2011, Qingdao, China, May 30–June 3, 2011, proceedings, edited by Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing, Lecture Notes Computer Science 6639, Springer, 2011. ISBN 978-3-642-20900-0.*, pages 62–80, 2011. (Cited on page 3.)
11. Daniel J. Bernstein. Simplified high-speed high-distance list decoding for alternant codes. In *Post-Quantum Cryptography 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 December 2, 2011, proceedings Lecture Notes Computer Science 7071. Springer.*, pages 200–216, 2011. (Cited on page 3.)
12. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: ball-collision decoding. In *Proceedings of the 31st annual conference on Advances in cryptology*, CRYPTO'11, pages 743–760, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited on pages 3, 12, and 13.)
13. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Wild McEliece incognito. In *Post-Quantum Cryptography, Fourth international workshop, PQCrypto 2011, Lecture Notes Computer Science 7071, Springer.*, pages 244–254, 2011. (Cited on page 3.)
14. Pierre-Louis Cayrel, Ayoub Otmani, and Damien Vergnaud. On Kabatianskii-Krouk-Smeets signatures. In *International Workshop on the Arithmetic of Finite Fields, WAIFI 2007, Springer, Lecture Notes Computer Science*, volume 4547, pages 237–251, 2007. (Cited on page 3.)
15. H. Chernoff. Asymptotic efficiency for tests based on the sum of observations. *Ann. Math. Stat.*, 23:493–507, 1952. (Cited on page 6.)
16. Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a McEliece-based digital signature scheme. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 157–174. Springer, 2001. (Cited on pages 3 and 5.)
17. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, November 1976. (Cited on page 3.)

18. P. Elias. List decoding for noisy channels, technical report 335. Technical report, Research Laboratory of Electronics, MIT, 1957. (Cited on pages 3 and 6.)
19. Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1–3):61–88, June 1999. (Cited on page 15.)
20. Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In *Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '09, pages 88–105, Berlin, Heidelberg, 2009. Springer-Verlag. (Cited on pages 3, 12, and 13.)
21. Ernst M. Gabidulin, A. V. Paramonov, and O. V. Tretjakov. Ideals over a non-commutative ring and their applications to cryptography. In D. W. Davies, editor, *Advances cryptology-EUROCRYPT '91. Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques held Brighton, April 8-11, Lecture Notes Computer Science 547. Springer ISBN 3-540-54620-0*, pages 482–489, 1991. (Cited on page 3.)
22. Philippe Gaborit. Shorter keys for code based cryptography. In *WCC 2005, Oyvind Ytrehus, Springer, Lecture Notes Computer Science, volume 3969*, pages 81–90, 2005. (Cited on page 3.)
23. Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. In *FOCS*, pages 28–39. IEEE Computer Society, 1998. (Cited on pages 3 and 6.)
24. Omessaad Hamdi, Sami Harari, and Ammar Bouallegue. Secure and fast digital signatures using BCH codes. *International Journal of Computer Science and Network Security*, 6(10):220–226, 2006. (Cited on page 3.)
25. L. Harn and D. C. Wang. Cryptanalysis and modification of digital signature scheme based on error-correcting codes. In *Electronics Letters 28*, pages 157–159, 1992. (Cited on page 3.)
26. Heeralal Janwa and Oscar Moreno. McEliece public key cryptosystems using algebraic-geometric codes. In *Designs, Codes and Cryptography 8*, pages 293–307, 1996. (Cited on page 3.)
27. S. M. Johnson. A new upper bound for error-correcting codes. *IRE Transactions on Information Theory*, IT-8:203–207, 1962. (Cited on page 6.)
28. Gregory Kabatianskii, E. Krouk, and Ben Smeets. A digital signature scheme based on random error-correcting codes. In Michael Darnell, editor, *Cryptography and coding. Proceedings of the 6th IMA International Conference held at the Royal Agricultural College, Cirencester, December 17-19, Lecture Notes Computer Science 1355. Springer*, pages 161–177, 1997. (Cited on page 3.)
29. G. Landais and J.P. Tillich. An efficient attack of a McEliece cryptosystem variant based on convolutional codes. In P. Gaborit, editor, *PQCrypto 2013*, volume 7932 of *LNCS*, pages 102–117. Springer, June 2013. (Cited on pages 3 and 18.)
30. P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In *Lecture Notes in Computer Science on Advances in Cryptology-EUROCRYPT'88*, pages 275–280, New York, NY, USA, 1988. Springer-Verlag New York, Inc. (Cited on pages 3, 12, and 13.)
31. J. S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Trans. Inf. Theor.*, 34(5):1354–1359, September 2006. (Cited on page 12.)
32. Yuan Xing Li and Chuanjia Liang. A digital signature scheme constructed with error-correcting codes. In *Chinese : Acta Electronica Sinica 19*, pages 102–104, 1991. (Cited on page 3.)
33. C. Löndahl and T. Johansson. A new version of McEliece PKC based on convolutional codes. In *ICICS*, pages 461–470, 2012. (Cited on pages 3 and 18.)
34. MAGMA. High performance software for algebra, number theory, and geometry — a large commercial software package. (Cited on page 15.)
35. Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $\tilde{\jmath}(2^{0.054n})$. In *Proceedings of the 17th international conference on The Theory and Application of Cryptology and Information Security*, ASIACRYPT'11, pages 107–124, Berlin, Heidelberg, 2011. Springer-Verlag. (Cited on pages 3, 12, and 13.)
36. R. J. McEliece. *A Public-Key System Based on Algebraic Coding Theory*, pages 114–116. Jet Propulsion Lab, 1978. DSN Progress Report 44. (Cited on page 3.)
37. R. Misoczki, J.-P. Tillich, N. Sendrier, and P.S.L.M. Barreto. MDPC-McEliece: New McEliece variants from Moderate Density Parity-Check codes. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 2069–2073, July 2013. (Cited on page 3.)
38. C. Monico, J. Rosenthal, and A. Shokrollahi. Using low density parity check codes in the McEliece cryptosystem. In *Information Theory, 2000. Proceedings. IEEE International Symposium on*, pages 215–, 2000. (Cited on page 3.)
39. Harald Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. In *Problems of Control and Information Theory 15*, pages 159–166, 1986. (Cited on page 3.)
40. A. Otmani and J.-P. Tillich. An efficient attack on all concrete KKS proposals. In *PQCrypto*, volume 7071 of *Lecture Notes in Computer Science*, pages 98–116, 2011. (Cited on page 3.)
41. E. Prange. The use of information sets in decoding cyclic codes. *IRE Transactions on Information Theory*, 8:5–9, 1962. (Cited on page 12.)
42. R.L. Rivest, A. Shamir, and L. Adelman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. (Cited on page 3.)

43. V.M. Sidelnikov. A public-key cryptosystem based on binary Reed-Muller codes. *Discrete Math. Appl.*, 4(3):1, 1994. (Cited on page 3.)
44. Jacques Stern. A method for finding codewords of small weight. In *Proceedings of the 3rd International Colloquium on Coding Theory and Applications*, pages 106–113, London, UK, UK, 1989. Springer-Verlag. (Cited on pages 3, 12, 13, and 17.)
45. Madhu Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13:180–193, 1997. (Cited on page 3.)
46. Johan van Tilburg. Cryptanalysis of the Alabbadi-Wicker digital signature scheme. In *Proceedings of Fourteenth Symposium on Information Theory in the Benelux*, pages 114–119, 1993. (Cited on page 3.)
47. Xinmei Wang. Digital signature scheme based on error-correcting codes. In *Electronics Letters*, volume 26, pages 898–899, 1990. (Cited on page 3.)
48. J. M. Wozencraft. List decoding. quarterly progress report. Technical report, Research Laboratory of Electronics, MIT, 1958. (Cited on pages 3 and 6.)
49. Sheng-Bo Xu, Jeroen Doumen, and Henk C. A. van Tilborg. On the security of digital signature schemes based on error-correcting codes. In *Designs, Codes and Cryptography*, volume 28, pages 187–199, 2003. (Cited on page 3.)

A Full Description of Sets of Parameters for Security Levels in the Range of $2^{80} - 2^{128}$

We denote with $K = (k_1, \ldots, k_w)$ and with $N = (n_1, \ldots, n_w)$ the vectors of values used in definition of concrete generator matrices as defined in equation (5). The used error set for all concrete codes is $E_2 = \{x \in \mathbb{F}_2^2 \mid wt(x) < 2\} = \{(0,0), (0,1), (1,0)\}$, thus $\ell = 2$. For computing the complexities of the rank attacks on the code and on its dual code we use equation (18), and we give here the values for K_t/ℓ and N_t/ℓ for which those complexities are achievable.

Codes for encryption

1. Code $(1160, 160)$.
   Public key size: 19.53 Kb.
   $w = 17$, $K = (32, 8, 8, \ldots, 8)$, $N = (32, 32, \ldots, 32, 488)$.
   Decoding complexity: $2^{25.36}$.
   Best rank attack complexity: 2^{90} for $K_t/\ell = 152$ and $N_t/\ell = 664$.
   The dual code is $(1160, 1000)$.
   Best dual code rank attack complexity: $2^{120.61}$ for $K_t/\ell = 968$ and $N_t/\ell = 1096$.

2. Code $(2050, 314)$.
   Public key size: 66.54 Kb.
   $w = 36$, $K = (34, 8, 8, \ldots, 8)$, $N = (32, 32, \ldots, 32, 616)$.
   Decoding complexity: $2^{26.94}$.
   Best rank attack complexity: $2^{113.75}$ for $K_t/\ell = 306$ and $N_t/\ell = 1426$.
   The dual code is $(2050, 1736)$.
   Best dual code rank attack complexity: $2^{121.20}$ for $K_t/\ell = 1704$ and $N_t/\ell = 1984$.

3. Code $(2980, 476)$.
   Public key size: 145.50 Kb.
   $w = 56$, $K = (36, 8, 8, \ldots, 8)$, $N = (32, 32, \ldots, 32, 744)$.
   Decoding complexity: $2^{28.53}$.
   Best rank attack complexity: $2^{131.96}$ for $K_t/\ell = 36$ and $N_t/\ell = 68$.
   The dual code is $(2980, 2504)$.
   Best dual code rank attack complexity: $2^{123.61}$ for $K_t/\ell = 2472$ and $N_t/\ell = 2912$.

4. Code $(3910, 638)$.
   Public key size: 254.83 Kb.
   $w = 76$, $K = (38, 8, 8, \ldots, 8)$, $N = (32, 32, \ldots, 32, 872)$.
   Decoding complexity: $2^{30.11}$.
   Best rank attack complexity: $2^{145.80}$ for $K_t/\ell = 38$ and $N_t/\ell = 70$.
   The dual code is $(3910, 3272)$.
   Best dual code rank attack complexity: $2^{126.44}$ for $K_t/\ell = 3240$ and $N_t/\ell = 3840$.

5. Code $(4840, 800)$.
   Public key size: 394.53 Kb.
   $w = 96$, $K = (40, 8, 8, \ldots, 8)$, $N = (32, 32, \ldots, 32, 1000)$.
   Decoding complexity: $2^{31.70}$.
   Best rank attack complexity: $2^{158.51}$ for $K_t/\ell = 40$ and $N_t/\ell = 72$.
   The dual code is $(4840, 4040)$.
   Best dual code rank attack complexity: $2^{129.37}$ for $K_t/\ell = 4008$ and $N_t/\ell = 4768$.

6. Code $(5730, 954)$.
   Public key size: 556.19 Kb.
   $w = 115$, $K = (42, 8, 8, \ldots, 8)$, $N = (32, 32, \ldots, 32, 1128)$.
   Decoding complexity: $2^{33.28}$.
   Best rank attack complexity: $2^{170.30}$ for $K_t/\ell = 42$ and $N_t/\ell = 74$.
   The dual code is $(5730, 4776)$.
   Best dual code rank attack complexity: $2^{132.36}$ for $K_t/\ell = 4744$ and $N_t/\ell = 5656$.

7. Code $(6660, 1116)$.
   Public key size: 755.26 Kb.
   $w = 135$, $K = (44, 8, 8, \ldots, 8)$, $N = (32, 32, \ldots, 32, 1256)$.
   Decoding complexity: $2^{34.86}$.
   Best rank attack complexity: $2^{181.88}$ for $K_t/\ell = 44$ and $N_t/\ell = 76$.
   The dual code is $(6660, 5544)$.
   Best dual code rank attack complexity: $2^{135.27}$ for $K_t/\ell = 5512$ and $N_t/\ell = 6584$.

8. Code $(7590, 1278)$.
   Public key size: 984.71 Kb.
   $w = 155$, $K = (46, 8, 8, \ldots, 8)$, $N = (32, 32, \ldots, 32, 1384)$.
   Decoding complexity: $2^{36.45}$.
   Best rank attack complexity: $2^{193.15}$ for $K_t/\ell = 46$ and $N_t/\ell = 78$.
   The dual code is $(7590, 6312)$.
   Best dual code rank attack complexity: $2^{138.17}$ for $K_t/\ell = 6280$ and $N_t/\ell = 7512$.

Codes for signatures

1. Code $(650, 306)$.
   Public key size: 12.85 Kb.
   $w = 6$, $K = (84, 48, 48, 48, 48, 30)$, $N = (48, 48, 48, 48, 48, 104)$.
   Decoding complexity: $2^{23.77}$.
   Best rank attack complexity: $2^{87.54}$ for $K_t/\ell = 276$ and $N_t/\ell = 516$.
   The dual code is $(650, 344)$.
   Best dual code rank attack complexity: $2^{93.32}$ for $K_t/\ell = 296$ and $N_t/\ell = 518$.

2. Code $(766, 366)$.
   Public key size: 17.87 Kb.
   $w = 7$, $K = (94, 48, \ldots, 48, 32)$, $N = (48, 48, \ldots, 48, 112)$.
   Decoding complexity: $2^{25.36}$.
   Best rank attack complexity: $2^{94.44}$ for $K_t/\ell = 334$ and $N_t/\ell = 622$.
   The dual code is $(766, 400)$.
   Best dual code rank attack complexity: $2^{98.93}$ for $K_t/\ell = 352$ and $N_t/\ell = 624$.

3. Code $(882, 426)$.
   Public key size: 23.71 Kb.
   $w = 8$, $K = (104, 48, \ldots, 48, 34)$, $N = (48, 48, \ldots, 48, 120)$.
   Decoding complexity: $2^{26.94}$.
   Best rank attack complexity: $2^{101.00}$ for $K_t/\ell = 392$ and $N_t/\ell = 728$.
   The dual code is $(882, 456)$.
   Best dual code rank attack complexity: $2^{104.48}$ for $K_t/\ell = 408$ and $N_t/\ell = 730$.

4. Code $(998, 486)$.
   Public key size: 30.37 Kb.
   $w = 9$, $K = (114, 48, \ldots, 48, 36)$, $N = (48, 48, \ldots, 48, 128)$.
   Decoding complexity: $2^{28.53}$.
   Best rank attack complexity: $2^{107.36}$ for $K_t/\ell = 450$ and $N_t/\ell = 834$.
   The dual code is $(998, 512)$.
   Best dual code rank attack complexity: $2^{110.00}$ for $K_t/\ell = 464$ and $N_t/\ell = 836$.

5. Code $(1114, 546)$.
   Public key size: 37.86 Kb.
   $w = 10$, $K = (124, 48, \ldots, 48, 38)$, $N = (48, 48, \ldots, 48, 136)$.
   Decoding complexity: $2^{30.11}$.
   Best rank attack complexity: $2^{113.55}$ for $K_t/\ell = 508$ and $N_t/\ell = 940$.
   The dual code is $(1114, 568)$.
   Best dual code rank attack complexity: $2^{115.48}$ for $K_t/\ell = 520$ and $N_t/\ell = 942$.

6. Code $(1230, 606)$.
   Public key size: 46.16 Kb.
   $w = 11$, $K = (134, 48, \ldots, 48, 40)$, $N = (48, 48, \ldots, 48, 144)$.
   Decoding complexity: $2^{31.69}$.
   Best rank attack complexity: $2^{119.63}$ for $K_t/\ell = 566$ and $N_t/\ell = 1046$.
   The dual code is $(1230, 624)$.
   Best dual code rank attack complexity: $2^{120.93}$ for $K_t/\ell = 576$ and $N_t/\ell = 1048$.

7. Code $(1346, 666)$.
   Public key size: 55.28 Kb.
   $w = 12$, $K = (144, 48, \ldots, 48, 42)$, $N = (48, 48, \ldots, 48, 152)$.
   Decoding complexity: $2^{33.28}$.
   Best rank attack complexity: $2^{125.61}$ for $K_t/\ell = 624$ and $N_t/\ell = 1152$.
   The dual code is $(1346, 680)$.
   Best dual code rank attack complexity: $2^{126.34}$ for $K_t/\ell = 632$ and $N_t/\ell = 1154$.

8. Code $(1462, 726)$.
   Public key size: 65.23 Kb.
   $w = 13$, $K = (154, 48, \ldots, 48, 44)$, $N = (48, 48, \ldots, 48, 160)$.
   Decoding complexity: $2^{34.87}$.
   Best rank attack complexity: $2^{131.52}$ for $K_t/\ell = 682$ and $N_t/\ell = 1258$.
   The dual code is $(1462, 736)$.
   Best dual code rank attack complexity: $2^{131.74}$ for $K_t/\ell = 688$ and $N_t/\ell = 1260$.

9. Code $(1578, 786)$.
   Public key size: 75.99 Kb.
   $w = 14$, $K = (164, 48, \ldots, 48, 46)$, $N = (48, 48, \ldots, 48, 168)$.
   Decoding complexity: $2^{36.45}$.
   Best rank attack complexity: $2^{137.37}$ for $K_t/\ell = 740$ and $N_t/\ell = 1364$.
   The dual code is $(1578, 792)$.
   Best dual code rank attack complexity: $2^{137.11}$ for $K_t/\ell = 744$ and $N_t/\ell = 1366$.