Probabilistic Analysis of LLL-based Decoder of Interleaved Chinese Remainder Codes

Matteo Abbondati, Antoine Afflatet, Eleonora Guerrini, Romain Lebreton

Probabilistic Analysis of LLL-based Decoder of Interleaved Chinese Remainder Codes

Matteo Abbondati, Antoine Afflatet, Eleonora Guerrini and Romain Lebreton
LIRMM, U. Montpellier, CNRS
Montpellier, France, FR

***Abstract*— To date, Li *et al.* have presented the only decoder for Interleaved Chinese Remainder (ICR) codes [1]. The core of their ICR decoder is to find a short vector in a lattice using the LLL algorithm [2]. However, their analysis of the decoding failure is partially heuristic. In this work, we present a new analysis of their LLL-based decoder that gives a proved upper bound on its decoding failure probability.** [1]

I. Introduction

Chinese Remainder Codes (CR codes) have been introduced for their theoretical and practical aspects. For instance, Goldreich *et al.* have shown their interest in secret sharing protocols [3]. CR codes are the integer counterpart of Reed-Solomon codes. They share similar correction capability and decoders.

Interleaving techniques are used to improve the unique decoding radius of a given code at the price of possible decoding failure. The most famous example is the case of Interleaved Reed Solomon (IRS) codes, which can be decoded asymptotically up to the Shannon bound with efficient (polynomial) decoders for almost all errors [4]. One can find in the literature a decoder for Interleaved Chinese Remainder codes that works in a similar way [1]. In both cases, the decoder tries to simultaneously correct ℓ codewords and build a system of linear equations, seeking for solutions satisfying some size constraints. The general idea of interleaved codes is that increasing ℓ means increasing the number of errors we can correct. But correcting beyond half the minimum distance of the code can lead to decoding failure. Thus, we will focus on the decoding failure probability with respect to the number of the errors we can correct.

The main difference between the polynomial decoder of IRS codes and the integer decoder of ICR codes is that the problem of finding the smallest degree solution becomes the problem of finding the shortest non-zero vector in a lattice, which is an NP-hard problem [5]. In the original paper of [1], the authors present the idea of using LLL for the decoding of ICR codes. However, the proof of the correctness of their decoder is justified by heuristic arguments. Note that the use of LLL already appeared in earlier work for the power decoding of CR codes [6].

In this paper, we prove an upper-bound on the decoding failure probability of the LLL-based decoder. Our main result expresses precisely the decoding failure of the algorithm in terms of its error correction capability (see Theorem 3.5). We consider this as a first step towards possible extensions of ICR codes to rational codes, and application to the design of a fault-tolerant integer linear system solver, similarly to the existing work in the polynomial case [7], [8].

The paper is structured as follows: in Section II we rephrase the presentation of Chinese Remainder codes and their decoder, in such a way that can be easily extended to ICR codes. Section III introduces ICR codes, describes their decoder and states our main result on the decoder correctness. We deal with the decoding failure analysis in Section IV. In Section V, we discuss our decoding failure probability on an example of code parameters.

II. Chinese Remainder codes

We start by recalling the CR codes and their decoding up to unique decoding radius.

We will denote $\mathbb{Z}_m$ the set of integers modulo m. The remainder of the Euclidean division of k by m is denoted indifferently $[k]_m$ or $(k \text{ rem } m)$.

Definition 2.1 (Chinese Remainder Codes): Let n be a positive integer, $p_1 < p_2 < \cdots < p_n$ be a list P of n distinct prime numbers. Let $N := \prod_{i=1}^{n} p_i$ be their products, and $K := \prod_{i=1}^{k} p_i$ for a parameter $1 \leq k \leq n$. The Chinese remainder code $CR(P; n, K)$ is a polyalphabetic code in the ambient space $\mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_n}$ of size N defined by

$$CR(P; n, K) = \{([C]_{p_1}, \ldots, [C]_{p_n}) \text{ s.t. } C \in \mathbb{N} \text{ and } C < K\}.$$

We notice that if $k = n$ our code does not have any correction capability, thus in what follows we will always tacitly assume that $k < n$.

Throughout this paper, we will denote with $\mathcal{C}$ a code $CR(P; n, K)$. For the transmission of a codeword c over a noisy channel, we write the received word $r = c + e$, where $e \in \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_n}$ is the error vector. According to the Chinese remainder theorem, we can associate a unique $R \in \mathbb{Z}_N$ to any word $r = (r_1, \ldots, r_n) \in \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_n}$. For the sake of simplicity, we will mostly take the point of view that the ambient space is $\mathbb{Z}_N$.

The Chinese Remainder codes, being polyalphabetic, leads us to the introduction of the weighted Hamming distance in order to take into account the various alphabets.

Definition 2.2 (Weighted Hamming distance): Let $R_1, R_2 \in \mathbb{Z}_N$, we define the *error support* $\mathcal{E}_{R_1,R_2}$ between R_1 and R_2 as the subset of indices i such that $R_1 \neq R_2 \bmod p_i$.

[1]This research work is supported by ANR-21-CE39-0009-BARRACUDA and ANR-21-CE39-0006-SANGRIA.

The *error locator* $\Lambda_{R_1,R_2} := \prod_{i\in\mathcal{E}_{R_1,R_2}} p_i$ is the product of the corresponding primes. Its complement is the *truth locator* $Y_{R_1,R_2} := N/\Lambda_{R_1,R_2} = \prod_{i\notin\mathcal{E}_{R_1,R_2}} p_i$. Finally, the *weighted Hamming distance* between R_1 and R_2 is defined as

$$d(R_1,R_2) = \log(\Lambda_{R_1,R_2}) = \sum_{i\in\mathcal{E}_{R_1,R_2}} \log(p_i).$$

In this paper, the logarithms are in base 2 by convention.

A. Decoding up to the unique decoding radius

We can now recall the minimum distance of a CR code and the classical decoder up to unique decoding radius.

Lemma 2.3 (Distance of a CR code): The minimal distance $d(\mathcal{C})$ of a $CR(P;n,K)$ code $\mathcal{C}$ satisfies $d(\mathcal{C}) > \log(\frac{N}{K})$.

Proof: For any $C_1, C_2 \in \mathcal{C}$ with $0 \le C_1 \le C_2 < K$, we have that $C := C_2 - C_1$ belongs to $\mathcal{C}$ and $d(C_1,C_2) = d(0,C)$. Now $d(0,C) = \log(\Lambda_{0,C}) = \log(N/Y_{0,C})$. Since $Y_{0,C}$ divides C and $C < K$, then we have $d(C_1,C_2) = d(0,C) > \log(\frac{N}{K})$. This inequality holds for any pair of codewords C_1, C_2, which implies that $d(\mathcal{C}) > \log(\frac{N}{K})$ as stated. ∎

For the rest of this section, let us fix a codeword $C \in \mathcal{C}$ and consider a received word $R \in \mathbb{Z}_N$ such that

$$d(C,R) \le d_u := \log(N/K)/2 = \log(\sqrt{N/K}). \quad (1)$$

We will use the simplified notations $\mathcal{E} := \mathcal{E}_{C,R}$, $Y := Y_{C,R}$ and $\Lambda := \Lambda_{C,R}$ for the rest of this section.

Since $d_u < d(\mathcal{C})/2$, unique decoding is possible ; we restate in Algorithm 1 the unique decoding algorithm described by Goldreich *et al.* [6], which is reminiscent of the Berlekamp-Welch decoder for Reed-Solomon codes [9].

Let $E = R - C \in \mathbb{Z}_N$ be the corresponding error. The truth locator Y divides E because $E = R - C = 0 \bmod p_i$ for all $i \notin \mathcal{E}$. Therefore, $N = \Lambda Y$ divides ΛE, which gives $\Lambda R = \Lambda E + \Lambda C = \Lambda C \bmod N$. In order to decode, one considers the following linearized equation in the unknowns (φ,ψ)

$$\varphi R = \psi \bmod N, \quad (2)$$

which admits $(\Lambda, \Lambda C)$ as a solution. Note that this solution is small: $\Lambda \le \sqrt{N/K}$ so that $\Lambda C < \sqrt{NK}$. Therefore, $(\Lambda, \Lambda C)$ actually belongs to

$$S_{\boldsymbol{R}} = \left\{ (\varphi,\psi) \in \mathbb{Z}^2 \;\middle|\; \begin{array}{c} \varphi R = \psi \bmod N \\ 0 \le \varphi \le \sqrt{N/K}, 0 \le \psi < \sqrt{NK} \end{array} \right\}.$$

Finding a non-zero element in $S_{\boldsymbol{R}}$ is a classical computer algebra problem named rational reconstruction (see *e.g.* [10, Section 5.10]). Using such a rational reconstruction algorithm, we can prove that Algorithm 1 is a decoder of CR codes.

Theorem 2.4: Algorithm 1 is correct, meaning that if R is within distance $d_u = \log\left(\sqrt{\frac{N}{K}}\right)$ of a codeword C, then it returns C. Conversely, if Algorithm 1 returns C, then C is a codeword such that $d(C,R) \le d_u$.

As a consequence, Algorithm 1 returns "decoding failure" if and only if there is no codeword within distance d_u of R.

Proof: Let us first assume that there exists a codeword C such that $d(C,R) \le d_u$. Let us show that the elements of $S_{\boldsymbol{R}}$ are unique as rationals, meaning that if (φ,ψ) and $(\varphi',\psi') \in S_{\boldsymbol{R}} \setminus \{(0,0)\}$, then $\psi/\varphi = \psi'/\varphi'$. Indeed, combining the equations $\varphi R = \psi \bmod N$ and $\varphi' R = \psi' \bmod N$, we obtain $\varphi'\psi = \varphi\psi' \bmod N$. We know that $0 \le \varphi'\psi < N$ and $0 \le \varphi\psi' < N$, so $\varphi'\psi = \varphi\psi'$.

Algorithm 1: Chinese Remainder codes decoder

Input: a code $CR(P;n,K)$ and a received word $R \in \mathbb{Z}_N$

Output: a codeword C such that $d(C,R) \le \log\left(\sqrt{\frac{N}{K}}\right)$ or "decoding failure"

```
Compute (φ, ψ) ∈ S_R by rational reconstruction
if φ divides ψ and 0 ≤ ψ/φ < K then
    return C := ψ/φ
else
    return "decoding failure"
```

We have already seen that $(\Lambda, \Lambda C) \in S_{\boldsymbol{R}}$. So the rational reconstruction $(\varphi,\psi) \in S_{\boldsymbol{R}} \setminus \{(0,0)\}$ computed by Algorithm 1 verifies that $\psi/\varphi = C$. In particular, φ divides ψ, $0 \le \psi/\varphi < K$ and Algorithm 1 returns C.

Next, we assume that Algorithm 1 returns $C := \psi/\varphi$ with $(\varphi,\psi) \in S_{\boldsymbol{R}} \setminus \{(0,0)\}$ and $0 \le \psi/\varphi < K$. This last inequality tells us that C is a codeword. Moreover, $(\varphi,\psi) \in S_{\boldsymbol{R}}$ implies that $\varphi(R-C) = 0 \bmod N$. So for any $i \in \mathcal{E}_{C,R}$, $p_i | \varphi(R-C)$ but $p_i \nmid (R-C)$, since $r_i \ne c_i$. This implies that $p_i | \varphi$. Altogether, we get that $\Lambda_{C,R}$ divides φ. Since $(\varphi,\psi) \in S_{\boldsymbol{R}}$, we have that $0 \le \varphi \le \sqrt{N/K}$ and we can conclude that $d(C,R) = \log(\Lambda_{C,R}) \le \log(\varphi) \le d_u$. ∎

III. Interleaved Chinese Remainder codes

Interleaving is a well-known encoding method giving efficient decoders for correcting a large amount of errors.

Definition 3.1 (Interleaved Chinese Remainder codes): Consider a CR code $\mathcal{C} := CR(P;n,K)$. A homogeneous interleaved Chinese remainder code $ICR(P;n,K)$ is defined as the set of matrices $\boldsymbol{C} = (c_{i,j})_{\substack{1\le i\le n \\ 1\le j\le \ell}}$ where $\boldsymbol{C}_{*,j} := (c_{1,j},\ldots,c_{n,j}) \in \mathcal{C}$ are codewords in the ambient space $\mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_n}$ for any $j = 1,\ldots,\ell$.

As with the CR codes, we can use the Chinese remainder theorem to see the ambient space as $(\mathbb{Z}_N)^\ell$. Therefore, we will write any received word $\boldsymbol{R}$ as a vector $\boldsymbol{R} = (R_1,\ldots,R_\ell) \in (\mathbb{Z}_N)^\ell$. In particular, we can write $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E}$ for an error $\boldsymbol{E} \in (\mathbb{Z}_N)^\ell$. We will now adapt the distance to ICR codes.

Definition 3.2 (Weighted Hamming distance for ICR codes): Let $\boldsymbol{R}, \boldsymbol{R}' \in (\mathbb{Z}_N)^\ell$, we define the *error support* $\mathcal{E}_{\boldsymbol{R},\boldsymbol{R}'}$ as the union $\cup_{j=1}^{\ell} \mathcal{E}_{R_j,R'_j}$ of error supports of CR codewords R_j and R'_j. In other words, the error support is the set of row indices i such that the corresponding rows differ $\boldsymbol{R}_{i,*} \ne \boldsymbol{R}'_{i,*}$. The *error locator* $\Lambda_{\boldsymbol{R},\boldsymbol{R}'} := \prod_{i\in\mathcal{E}_{\boldsymbol{R},\boldsymbol{R}'}} p_i$ is the product of the corresponding primes. Its complement is the *truth locator* $Y_{\boldsymbol{R},\boldsymbol{R}'} := N/\Lambda_{\boldsymbol{R},\boldsymbol{R}'}$. The *weighted Hamming distance* between $\boldsymbol{R}$ and $\boldsymbol{R}'$ is defined as $d(\boldsymbol{R},\boldsymbol{R}') := \log(\Lambda_{\boldsymbol{R},\boldsymbol{R}'})$.

In order to decode ICR codes, we can set the following key equations similarly to CR codes:

$$\varphi R_j = \psi_j \bmod N \quad \text{for all } 1 \le j \le \ell. \tag{3}$$

Note that $v_{\boldsymbol{C}} := (\Lambda, \Lambda C_1, \ldots, \Lambda C_\ell)$ is a solution where $\Lambda := \Lambda_{\boldsymbol{C},\boldsymbol{R}}$.

One could decode an ICR code by solving each one of Equations (3) separately using Algorithm 1. However, by solving the equations in this way, one can only decode below the unique decoding capacity d_u introduced in Equation (1). In order to decode beyond this quantity we consider a collaborative decoder that simultaneously decodes the ℓ codewords.

We follow the setting of Li *et al.* [1], which propose an approach similar to the simultaneous decoder of Interleaving Reed Solomon codes in [11]. Reed Solomon codes can be viewed as the polynomial version of CR codes, and IRS decoding can be performed finding a polynomial vector of small row-degree of a $\mathbb{F}_q[x]-$module. Intuitively, since for CR codes we have Theorem 2.4, we can transpose this to the interleaved case, dealing with the problem of finding a vector of integers of small size in a lattice (*i.e.* a $\mathbb{Z}-$module).

The solutions of Equations (3) form a lattice. Recall that the *lattice* generated by $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n$ is defined as $\mathcal{L} := \{\sum_{1\le i\le n} k_i \boldsymbol{f}_i : k_1, \ldots, k_n \in \mathbb{Z}\}$ where $n \in \mathbb{N}$ and $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_n \in \mathbb{R}^n$. From Equations (3), there exist $m_1, \ldots, m_\ell \in \mathbb{Z}$ such that $(\varphi, \psi_1, \ldots, \psi_\ell) = (\varphi, \varphi R_1 + m_1 N, \ldots, \varphi R_\ell + m_\ell N)$. So the set of solutions of Equations (3) is the lattice $\mathcal{L}$ generated by the rows of the following matrix

$$\left(\begin{array}{c|c} 1 & \boldsymbol{R} \\ \hline \boldsymbol{0} & N \cdot \mathrm{Id}_\ell \end{array}\right) \in \mathbb{Z}^{(\ell+1)\times(\ell+1)}. \tag{4}$$

In particular, $v_{\boldsymbol{C}} \in \mathcal{L}$.

A. Decoding using a short vector

As for decoding CR codes, the target solution $v_{\boldsymbol{C}}$ has small size, so we redefine $S_{\boldsymbol{R}}$ as a subset of solutions of Equations (3) with size constraints

$$S_{\boldsymbol{R}} = \left\{ (\varphi, \psi_1, \ldots, \psi_\ell) \in \mathbb{Z}^{\ell+1} \;\middle|\; \begin{array}{c} \varphi R_i = \psi_i \bmod N \\ |\varphi| \le 2^\tau,\ |\psi_i| < 2^\tau K \end{array} \right\}$$

for some distance parameter τ that we will soon discuss. Then, decoding ICR codes can be reduced to the problem of finding the shortest non-zero vector of a lattice, which is proved NP-hard [5]. Thus, we deal with a well-known approximation algorithm, named LLL [2]. Given a lattice $\mathcal{L}$, the LLL algorithm finds, in polynomial time, a vector v_s such that $||v_s||_2 \le \gamma ||\lambda_1(\mathcal{L})||_2$, where $\lambda_1(\mathcal{L})$ is the shortest non-zero vector of the lattice and $\gamma \ge 1$ is the approximation constant of LLL. It is proved that LLL always outputs a γ-approximation of the shortest vector for the value $\gamma = \sqrt{2}^\ell$ (our lattice has dimension $\ell + 1$). Note that experiments on LLL and its variants show that it achieves $\gamma \approx 1.02^{\ell+1}$ on average [12].

Remark that the output of LLL has constraints on the 2-norm $||\,.\,||_2$. However, a short vector for the 2-norm may not belong to $S_{\boldsymbol{R}}$ for a small τ. In order to make the two notions of size match, we need to introduce a scaling operator $\sigma_K : \mathbb{Q}^{\ell+1} \to \mathbb{Q}^{\ell+1}$ such that $\sigma_K((v_0, v_1, \ldots, v_\ell)) = (v_0 K, v_1, \ldots, v_\ell)$. We can then define $\bar{\mathcal{L}} := \sigma_K(\mathcal{L})$ and $v_s := \sigma_K^{-1}(\bar{v}_s)$, where $\bar{v}_s := \mathrm{LLL}(\bar{\mathcal{L}})$ is a short vector. We can now prove our result, which requires the following condition on $\boldsymbol{R}$.

Constraint 1: There exists a codeword $\boldsymbol{C}$ such that $\gamma\sqrt{\ell+1}\Lambda_{\boldsymbol{C},\boldsymbol{R}} \le 2^\tau$.

This constraint is equivalent to $d(\boldsymbol{C}, \boldsymbol{R}) \le \tau - \log(\gamma\sqrt{\ell+1})$.

Lemma 3.3: Assuming Constraint 1, we have that $v_s \in S_{\boldsymbol{R}}$.

Proof: Let $\bar{v}_{\boldsymbol{C}} := \sigma_K(v_{\boldsymbol{C}})$ belongs to $\bar{\mathcal{L}}$. Because LLL computes a γ-approximation of the shortest vector, we have $||\bar{v}_s||_2 \le \gamma ||\lambda_1(\bar{\mathcal{L}})||_2 \le \gamma ||\bar{v}_{\boldsymbol{C}}||_2 \le \gamma\sqrt{\ell+1}\Lambda K$, where $\lambda_1(\bar{\mathcal{L}})$ is the actual shortest non-zero vector of $\bar{\mathcal{L}}$. Note that $\bar{v}_s \in \bar{\mathcal{L}}$ implies $v_s \in \mathcal{L}$. Moreover, $||\bar{v}_s||_\infty \le ||\bar{v}_s||_2 < \gamma\sqrt{\ell+1}\Lambda K \le 2^\tau K$ using Constraint 1. So if we denote $v_s = (\varphi, \psi_1, \ldots, \psi_\ell)$, then $|\varphi| \le 2^\tau$ and $|\psi_j| < 2^\tau K$ for $1 \le j \le \ell$. Altogether, we have proved that $v_s \in S_{\boldsymbol{R}}$. ∎

Note that, assuming Constraint 1, we also have that $v_{\boldsymbol{C}} = (\Lambda, \Lambda C_1, \ldots, \Lambda C_\ell)$ belongs to $S_{\boldsymbol{R}}$. We have now all the ingredients to present our slightly modified version of the ICR codes decoder of [1].

Algorithm 2: Interleaved CR codes decoder

Input: $ICR(P; n, K)$, received word $\boldsymbol{R}$, parameter τ
Output: A codeword $\boldsymbol{C}$ s.t. $d(\boldsymbol{C}, \boldsymbol{R}) \le \tau$ or "decoding failure"

```
Let L̄ be the scaled lattice
Compute a short vector v̄_s := LLL(L̄)
Unscale the vector: v_s = (φ, ψ_1, ..., ψ_ℓ) := σ_K^{-1}(v̄_s)
if (|φ| ≤ 2^τ) and (φ divides all the (ψ_j)_{j=1,...,ℓ}) and
   (0 ≤ ψ_j/φ < K for all 1 ≤ j ≤ ℓ) then
    return (C_1, ..., C_ℓ) := (ψ_1/φ, ..., ψ_ℓ/φ)
else
    return "decoding failure"
end
```

Compared to Theorem 2.4 on CR codes, only one implication in the correctness of Algorithm 2 is always true (see upcoming Lemma 3.4). The other implication will be probabilistic; we will discuss it in the forthcoming Section III-B.

Lemma 3.4: If Algorithm 2 returns $\boldsymbol{C}$ on input $\boldsymbol{R}$ and parameter τ, then $\boldsymbol{C}$ is a codeword of $ICR(P; n, K)$ such that $d(\boldsymbol{C}, \boldsymbol{R}) \le \tau$.

Proof: $\boldsymbol{C}$ is a codeword of $ICR(P; n, K)$ because the algorithm has verified that $0 \le C_j = \psi_j/\varphi < K$. Now, since φ divides all the ψ_j, we have that $(\varphi, \psi_1, \ldots, \psi_\ell) = (\varphi, \varphi C_1, \ldots, \varphi C_\ell)$ so that $\varphi R_j = \varphi C_j \bmod N$ for all j. In particular, $p_i | \varphi(R_j - C_j)$ for all i and j. However, for all $i \in \mathcal{E}_{\boldsymbol{C},\boldsymbol{R}}$, there exists j such that $p_i \nmid (R_j - C_j)$, which implies that $p_i | \varphi$. As a consequence, $\Lambda_{\boldsymbol{C},\boldsymbol{R}} | \varphi$. Considering that $|\varphi| \le 2^\tau$, we can conclude that $d(\boldsymbol{C}, \boldsymbol{R}) = \log \Lambda_{\boldsymbol{C},\boldsymbol{R}} \le \log \varphi \le \tau$. ∎

B. Distribution of random received word

The decoder described in Algorithm 2 must occasionally fail, since it tries to uniquely decode the received word above the unique decoding capacity. In order to analyze our algorithm, we consider a random received word $\boldsymbol{R}$. Similarly to the literature on IRS [4], [13], [14], the distribution $\mathcal{D}_{\boldsymbol{C},\mathcal{E}_r}$ of $\boldsymbol{R}$ is related to a codeword $\boldsymbol{C}$ and a support for random errors $\mathcal{E}_r \subseteq \{1,\dots,n\}$. Then, the distribution $\mathcal{D}_{\boldsymbol{C},\mathcal{E}_r}$ is obtained by taking $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E}$ with $\boldsymbol{E}$ as follows: if $i \notin \mathcal{E}_r$, then $\boldsymbol{E}_{i,*} := (e_{i,j})_{1\le j\le \ell}$ is the zero vector. If $i \in \mathcal{E}_r$, then the ℓ entries of $\boldsymbol{E}_{i,*}$ are mutually independent random variables following the uniform distribution on $\mathbb{Z}_{p_i}$, which we will denote as $\boldsymbol{E}_{i,*} \sim \mathcal{U}_{\perp\!\!\!\perp}((\mathbb{Z}_{p_i})^\ell)$.

Let $Y_r := \prod_{i\notin\mathcal{E}_r} p_i$ (resp. $\Lambda_r := \prod_{i\in\mathcal{E}_r} p_i$) be the truth locator (resp. error locator) associated to $\mathcal{E}_r$. If we use the point of view where $\boldsymbol{R},\boldsymbol{C},\boldsymbol{E}$ all belong to $(\mathbb{Z}_N)^\ell$, then the distribution $\mathcal{D}_{\boldsymbol{C},\mathcal{E}_r}$ can be rephrased as taking $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E}$ with $\boldsymbol{E} = Y_r \cdot \boldsymbol{E}'$ and $\boldsymbol{E}' \sim \mathcal{U}_{\perp\!\!\!\perp}((\mathbb{Z}_{\Lambda_r})^\ell)$.

Theorem 3.5 (Main result): Given an ICR code $\mathcal{C}$ with parameters N, K, ℓ, and the approximation constant γ of LLL, set

$$d_{\max} := \frac{\ell}{\ell+1}\Big[\log(N/K) - \log(6\gamma\sqrt{\ell+1})\Big]. \tag{5}$$

Choose a decoding distance bound $d_t < d_{\max}$, and set the parameter $\tau_t := d_t + \log(\gamma\sqrt{\ell+1})$ in Algorithm 2. Consider a random received word $\boldsymbol{R} \sim \mathcal{D}_{\boldsymbol{C},\mathcal{E}_r}$, for some codeword $\boldsymbol{C}\in\mathcal{C}$ and error support $\mathcal{E}_r$ such that $\log\Lambda_r \le d_t$.

Then, Algorithm 2 on random input $\boldsymbol{R}$ outputs the center codeword $\boldsymbol{C}$ of the distribution $\mathcal{D}_{\boldsymbol{C},\mathcal{E}_r}$, with a probability of failure $\mathbb{P}_f$ upper-bounded by

$$\mathbb{P}_f \le 2^{-(\ell+1)(d_{\max}-d_t)} + \exp\left(\frac{n}{p_1^{\ell-1}}\right) - 1.$$

In other words, Theorem 3.5 states that Algorithm 2 can decode almost all received word $\boldsymbol{R}$ within distance $d_t < d_{\max}$.

IV. ICR Decoder Analysis

In this section, we analyze the correctness of Algorithm 2. For this matter, we will discuss the probability of the event $S_{\boldsymbol{R}} \subseteq \mathbb{Z}v_{\boldsymbol{C}}$. When this result is verified, then v_s is colinear to $v_{\boldsymbol{C}}$ and Algorithm 2 succeeds.

Throughout this section, we work under the assumptions of Theorem 3.5. In particular, we will consider the error model defined in Section III-B, *i.e.* that $\boldsymbol{R}$ is drawn at random from the distribution $\mathcal{D}_{\boldsymbol{C},\mathcal{E}_r}$ with $\log\Lambda_{\mathcal{E}_r} \le d_t$. For the analysis, we set $\tau \leftarrow \tau_t$ in $S_{\boldsymbol{R}}$, where τ_t is defined in Theorem 3.5.

A. Decoding failure probability

In order to have that the decoding algorithm succeeds, we will ensure that

$$S_{\boldsymbol{R}} \subseteq (\Lambda, \Lambda C_1, \dots, \Lambda C_\ell)\mathbb{Z}. \tag{6}$$

Recall that Λ is a short notation for $\Lambda_{\boldsymbol{C},\boldsymbol{R}}$, and also $\mathcal{E} := \mathcal{E}_{\boldsymbol{C},\boldsymbol{R}}$, $Y := Y_{\boldsymbol{C},\boldsymbol{R}}$. To study the condition (6), we need to introduce the set

$$S_{\boldsymbol{E}'} := \left\{\varphi \in \mathbb{Z}_{\Lambda_r} \;\middle|\; |\varphi E'_i \text{ crem } \Lambda_r| \le 2^{\tau_t+1}K\Lambda_r/N\right\}$$

where $e \text{ crem } m$ denotes the *central remainder* of e modulo m, that is the unique representative of $e \bmod m$ in the range $\{-\lceil\frac{m}{2}\rceil+1,\dots,\lfloor\frac{m}{2}\rfloor\}$. The central remainder has the property of being the representative with the smallest absolute value.

We need a new constraint to prove the following Lemma.

Constraint 2: $2^{\tau_t+1}K \le N$,

Lemma 4.1: Assume that Constraint 2 is verified. Then the inclusion $S_{\boldsymbol{E}'} \subseteq \Lambda(\mathbb{Z}_{\Lambda_r})$ implies $S_{\boldsymbol{R}} \subseteq (\Lambda, \Lambda C_1, \dots, \Lambda C_\ell)\mathbb{Z}$.

Proof: Let $(\varphi, \psi_1, \dots, \psi_\ell) \in S_{\boldsymbol{R}}$ then

$$\varphi E_i = \varphi(R_i - C_i) = \psi_i - \varphi C_i \bmod N. \tag{7}$$

Since $\boldsymbol{R}$ comes from the distribution $\mathcal{D}_{\boldsymbol{C},\mathcal{E}_r}$ related to the error support $\mathcal{E}_r$, we have that $\mathcal{E} \subseteq \mathcal{E}_r$, $\Lambda|\Lambda_r$, and so $Y_r|Y$. Since $Y|E_i$, we conclude that $Y_r|E_i$. Since $Y_r|N$ as well, we obtain that $\psi_i = \varphi C_i \bmod Y_r$. We can define the integer $\psi'_i := \frac{\psi_i - \varphi C_i}{Y_r}$, obtaining the bound $|\psi'_i| \le \frac{|\psi_i|+|\varphi C_i|}{Y_r} < \frac{2^{\tau_t+1}K}{N}\Lambda_r$ which, thanks to Constraint 2, leads to

$$|\psi'_i| < \Lambda_r. \tag{8}$$

Dividing every term in the equation (7) by Y_r we obtain

$$\varphi E'_i = \psi'_i \bmod \Lambda_r. \tag{9}$$

Thus, $\varphi \bmod \Lambda_r$ belongs to $S_{\boldsymbol{E}'}$, and thanks to the hypothesis, there exists a such that $\varphi = a\Lambda$. Since $\Lambda|\varphi$ and $Y|E_i$, we obtain that $\Lambda_r = (\Lambda Y)/Y_r$ divides $\varphi E'_i = (\varphi E_i)/Y_r$. Therefore, using Equation (9), we get $\psi'_i = 0 \bmod \Lambda_r$ and the inequality (8) implies that $\psi'_i = 0$.

Thus, we have that $\psi_i = \varphi C_i = a\Lambda C_i$, and we conclude that $(\varphi, \psi_1 \dots, \psi_\ell) \in (\Lambda, \Lambda C_1, \dots, \Lambda C_\ell)\mathbb{Z}$. ∎

Thanks to the previous lemma, we can compute an upper bound for the failure probability $\mathbb{P}_f$ of the decoding algorithm. Indeed, if Algorithm 2 fails, then necessarily the vector v_s obtained from LLL is not colinear to $v_{\boldsymbol{C}}$. Since $v_s \in S_{\boldsymbol{R}}$, this can happen only if $S_{\boldsymbol{R}} \not\subseteq v_{\boldsymbol{C}}\mathbb{Z}$. The previous lemma can be restated as

$$S_{\boldsymbol{R}} \not\subseteq v_{\boldsymbol{C}}\mathbb{Z} \Rightarrow S_{\boldsymbol{E}'} \not\subseteq \Lambda(\mathbb{Z}_{\Lambda_r}) \Rightarrow S_{\boldsymbol{E}'} \neq \{0\}.$$

Thus, we can say that the decoding failure probability $\mathbb{P}_f$ of Algorithm 2 satisfies $\mathbb{P}_f \le \mathbb{P}(S_{\boldsymbol{E}'} \neq \{0\})$. In order to analyze this probability, we introduce the notations $B := \frac{2^{\tau_t+1}K}{N}\Lambda_r$ and $g_\varphi = \gcd(\varphi, \Lambda_r)$ for $\varphi \in \mathbb{Z}_{\Lambda_r}$.

Lemma 4.2: Given $\boldsymbol{E}' \sim \mathcal{U}_{\perp\!\!\!\perp}((\mathbb{Z}_{\Lambda_r})^\ell)$, then

$$\mathbb{P}(S_{\boldsymbol{E}'} \neq \{0\}) \le \frac{1}{\Lambda_r^\ell}\sum_{\varphi=1}^{\Lambda_r-1}\left(2\left\lfloor\frac{B}{g_\varphi}\right\rfloor g_\varphi + g_\varphi\right)^\ell.$$

Proof: We will denote $\mathbb{Z}_{\Lambda_r}^{\neq 0} := \mathbb{Z}_{\Lambda_r} \setminus \{0\}$.

$$\begin{aligned}\mathbb{P}(S_{\boldsymbol{E}'} \neq \{0\}) &= \mathbb{P}\left(\bigcup_{\varphi\in\mathbb{Z}_{\Lambda_r}^{\neq 0}}\bigcap_{i=1}^{\ell}\{|\varphi E'_i \text{ crem } \Lambda_r| \le B\}\right)\\ &\le \sum_{\varphi\in\mathbb{Z}_{\Lambda_r}^{\neq 0}}\mathbb{P}\left(\bigcap_{i=1}^{\ell}\{|\varphi E'_i \text{ crem } \Lambda_r| \le B\}\right).\end{aligned}$$

Since the random variables $\{E'_i\}_{i=1}^{\ell}$ are mutually independent and share the same distribution, the multiples $\{\varphi E'_i\}_{i=1}^{\ell}$ are also mutually independent for every $\varphi \in \mathbb{Z}_{\Lambda_r}^{\neq 0}$. So, we can write

$$\mathbb{P}(S_{\boldsymbol{E}'} \neq \{0\}) \leq \sum_{\varphi=1}^{\Lambda_r - 1} \left(\mathbb{P}(|\varphi U \text{ crem } \Lambda_r| \leq B)\right)^{\ell}$$

with $U \sim \mathcal{U}(\mathbb{Z}_{\Lambda_r})$ being a uniform random variable on $\mathbb{Z}_{\Lambda_r}$. The distribution of φU is uniform on the orbit $\mathcal{O}_{\varphi}$ of φ

$$\mathcal{O}_{\varphi} = \left\{0, \varphi, 2\varphi, \ldots, \left(\frac{\Lambda_r}{g_{\varphi}} - 1\right)\varphi\right\} \subseteq \mathbb{Z}_{\Lambda_r}.$$

Since for every $\varphi \in \mathbb{Z}_{\Lambda_r}^{\neq 0}$ there exists a unique unit (invertible element) $u \in \mathbb{Z}_{\Lambda_r}^{*}$ such that $\varphi = g_{\varphi} u$ and u is a generator of $\mathbb{Z}_{\Lambda_r}$, *i.e.* $u\mathbb{Z}_{\Lambda_r} = \mathbb{Z}_{\Lambda_r}$, we can conclude that the orbit $\mathcal{O}_{\varphi}$ is equal to the orbit $\mathcal{O}_{g_{\varphi}}$. Therefore $\varphi U \sim \mathcal{U}(\mathcal{O}_{g_{\varphi}})$.

$$\mathbb{P}(|\varphi U \text{ crem } \Lambda_r| \leq B) = \frac{\#\{y \in \mathcal{O}_{g_{\varphi}} \text{ s.t. } |y \text{ crem } \Lambda_r| \leq B\}}{\#\mathcal{O}_{g_{\varphi}}}$$

$$\leq \frac{2\#\{1 \leq s \leq \frac{\Lambda_r}{g_{\varphi}} - 1 \mid g_{\varphi} s \leq B\} + 1}{\frac{\Lambda_r}{g_{\varphi}}} = \frac{g_{\varphi}\left(2\lfloor \frac{B}{g_{\varphi}} \rfloor + 1\right)}{\Lambda_r}.$$

The inequality comes from the fact that, within the orbit $\mathcal{O}_{g_{\varphi}}$, we count the remainders $\text{crem } \Lambda_r$ which are less than or equal to B by doubling the number of remainders $\bmod \Lambda_r$ between 1 and B and we add 1 to include zero. This latter way of counting can yield a higher result when $B > \frac{\Lambda_r}{2}$. ∎

We can obtain an explicit upper bound of the above formula for the failure probability in terms of the parameters of the code and of the error model.

Lemma 4.3:

$$\mathbb{P}(S_{\boldsymbol{E}'} \neq \{0\}) \leq \left(3\frac{2^{\tau_t+1}K}{N}\right)^{\ell} \Lambda_r + \exp\left(\frac{n}{p_1^{\ell-1}}\right) - 1. \quad (10)$$

Proof: Thanks to the previous lemma, we need to obtain an upper bound for the sum $\frac{1}{\Lambda_r^{\ell}} \sum_{\varphi=1}^{\Lambda_r - 1} \left(2\left\lfloor \frac{B}{g_{\varphi}} \right\rfloor g_{\varphi} + g_{\varphi}\right)^{\ell}$. As the sum is over all the elements $\varphi \in \mathbb{Z}_{\Lambda_r}^{\neq 0}$ but the generic term depends only on $g_{\varphi} = \gcd(\varphi, \Lambda_r)$, we can regroup the terms of the sum based on their gcd's with Λ_r. For every given divisor $d < \Lambda_r$ of Λ_r, the number of $\varphi \in \mathbb{Z}_{\Lambda_r}^{\neq 0}$ with $g_{\varphi} = d$ is equal to $\phi\left(\frac{\Lambda_r}{d}\right)$, where ϕ is the Euler's totient function. This is because $\{\varphi \in \mathbb{Z}_{\Lambda_r} \mid g_{\varphi} = d\} = d(\mathbb{Z}_{\Lambda_r/d})^*$ whose cardinality is $\phi\left(\frac{\Lambda_r}{d}\right)$. Thus we can write

$$\sum_{\varphi=1}^{\Lambda_r - 1} \left(2\left\lfloor \frac{B}{g_{\varphi}} \right\rfloor g_{\varphi} + g_{\varphi}\right)^{\ell} = \sum_{\substack{d \mid \Lambda_r \\ d \neq \Lambda_r}} \phi\left(\frac{\Lambda_r}{d}\right) \left(2\left\lfloor \frac{B}{d} \right\rfloor d + d\right)^{\ell}.$$

Now we distinguish two cases: if $d > B$ then $\lfloor \frac{B}{d} \rfloor = 0$ and the generic term of the sum reduces to $\phi\left(\frac{\Lambda_r}{d}\right)(d)^{\ell}$, while when $d \leq B$ we upper bound the generic term of the sum with $\phi\left(\frac{\Lambda_r}{d}\right)(3B)^{\ell}$. As we do not have direct control on the average number of divisors which are either in the first or in the second class, we will upper-bound the sum by a double sum considering both classes at the same time. Thus, our sum is upper bounded by the following:

$$\begin{aligned}
&\frac{1}{\Lambda_r^{\ell}} \sum_{\substack{d \mid \Lambda_r \\ d \neq \Lambda_r}} \phi\left(\frac{\Lambda_r}{d}\right)\left((3B)^{\ell} + d^{\ell}\right) \\
&= \left(\frac{3B}{\Lambda_r}\right)^{\ell} \sum_{\substack{d \mid \Lambda_r \\ d \neq \Lambda_r}} \phi\left(\frac{\Lambda_r}{d}\right) + \frac{1}{\Lambda_r^{\ell}} \sum_{\substack{d \mid \Lambda_r \\ d \neq \Lambda_r}} \phi\left(\frac{\Lambda_r}{d}\right) d^{\ell} \\
&\leq \left(\frac{3B}{\Lambda_r}\right)^{\ell} \Lambda_r + \frac{1}{\Lambda_r^{\ell-1}} \sum_{\substack{d \mid \Lambda_r \\ d \neq \Lambda_r}} d^{\ell-1}.
\end{aligned}$$

Studying the second sum, we recognize the divisor sum function $\sigma_{\ell-1}(\Lambda_r) := \sum_{d \mid \Lambda_r} d^{\ell-1}$. Observing that this arithmetic function is multiplicative, i.e. $\sigma_k(ab) = \sigma_k(a)\sigma_k(b)$ if $\gcd(a, b) = 1$, we can write

$$\begin{aligned}
\frac{1}{\Lambda_r^{\ell-1}} \sum_{d \mid \Lambda_r} d^{\ell-1} &= \frac{\prod_{i \in \mathcal{E}_r} \sigma_{\ell-1}(p_i)}{\prod_{i \in \mathcal{E}_r} p_i^{\ell-1}} = \prod_{i \in \mathcal{E}_r} \frac{1 + p_i^{\ell-1}}{p_i^{\ell-1}} \\
&= \prod_{i \in \mathcal{E}_r} \left(1 + \frac{1}{p_i^{\ell-1}}\right) \leq \left(1 + \frac{1}{p_1^{\ell-1}}\right)^{|\mathcal{E}_r|}
\end{aligned}$$

where in the last inequality we used that $p_1 < p_2 < \cdots < p_n$. Now to conclude we observe that

$$\begin{aligned}
\left(1 + \frac{1}{p_1^{\ell-1}}\right)^{|\mathcal{E}_r|} &= \exp\left(|\mathcal{E}_r| \ln\left(1 + \frac{1}{p_1^{\ell-1}}\right)\right) \\
&\leq \exp\left(|\mathcal{E}_r| \frac{1}{p_1^{\ell-1}}\right) \leq \exp\left(\frac{n}{p_1^{\ell-1}}\right)
\end{aligned}$$

where we used that $\ln(1 + x) \leq x$ for every $x \in \mathbb{R}$. ∎

B. Proof of the main Theorem

Proof of Theorem 3.5: We consider the execution of Algorithm 2 with the input parameter τ_t and for a random received word $\boldsymbol{R}$. Let us first verify that our choice of parameters satisfy Constraint 1, so that $v_s \in S_{\boldsymbol{R}}$. Indeed, $\Lambda \leq \Lambda_r \leq 2^{d_t} = 2^{\tau_t}/(\gamma\sqrt{\ell+1})$.

Next, we need to verify Constraint 2 so that we can apply Lemma 4.1, and deduce that the probability of failure of Algorithm 2 is upper-bounded by $\mathbb{P}_{\boldsymbol{E}'}(S_{\boldsymbol{E}'} \neq \{0\})$. So let us prove that $2^{\tau_t+1}K/N \leq 1$. Note first that

$$\begin{aligned}
2^{\tau_t+1}K/N &= 2(2^{d_t}\gamma\sqrt{\ell+1})K/N \\
&\leq 2^{d_{\max}}(2\gamma\sqrt{\ell+1}K/N) \\
&= \left(6\gamma\sqrt{\ell+1}K/N\right)^{-\ell/(\ell+1)}(2\gamma\sqrt{\ell+1}K/N) \\
&= \left(\frac{2K}{N}\frac{\gamma\sqrt{\ell+1}}{3^{\ell}}\right)^{1/(\ell+1)}.
\end{aligned}$$

We can assume $2K \leq N$ since we took $k < n$ after Definition 2.1. Moreover, the approximation constant γ of LLL always satisfies $\gamma \leq \sqrt{2}^{\ell}$. So, it is not hard to be convinced that $\gamma^{\ell}\sqrt{\ell+1} \leq \sqrt{2}^{\ell}\sqrt{\ell+1} \leq 3^{\ell}$ for every $\ell \in \mathbb{N}$. Altogether, we have proved $2^{\tau_t+1}K/N \leq 1$ and Constraint 2.

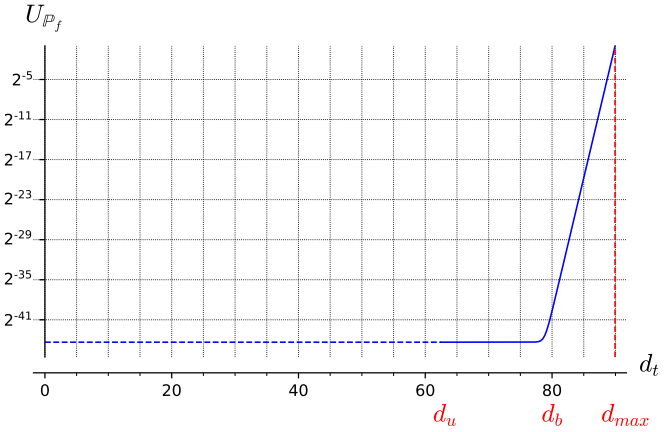
At this point, we can apply Lemma 4.3 to get an upper-bound on the failure probability. It remains to prove that $\left(3\frac{2^{\tau_t+1}K}{N}\right)^{\ell}\Lambda_r \leq 2^{-(\ell+1)(d_{\max}-d_t)}$. Since $2^{-d_{\max}(\ell+1)} = (6K\gamma\sqrt{\ell+1}/N)^{\ell}$, the previous inequality is equivalent to $\Lambda_r 2^{\tau_t \ell} \leq 2^{d_t(\ell+1)}(\gamma\sqrt{\ell+1})^{\ell}$, which is verified since $2^{\tau_t} = 2^{d_t}\gamma\sqrt{\ell+1}$ and $\Lambda_r \leq 2^{d_t}$. ∎

V. Discussion on the Failure Probability

Here we plot the upper bound on the failure probability

$$U_{\mathbb{P}_f}(d_t) := 2^{-(\ell+1)(d_{max}-d_t)} + \left(\exp\left(n/p_1^{\ell-1}\right) - 1\right) \quad (11)$$

from Theorem 3.5 as a function of the distance.



We used a logarithmic scale on the y-axis. This represents the situation in which we use $n = 50$ primes for the ICR code and each of them has about 25 bits. The other parameters used for the code are $k = 45$ and $\ell = 3$.

The graph exhibits a clear asymptotic behavior of the failure probability, which becomes predominant for distances less than a corner distance d_b.

This corner distance corresponds to the distance in which the first term in (11) is equal to the constant term of $U_{\mathbb{P}_f}$, *i.e.* d_b is such that $2^{-(\ell+1)(d_{max}-d_b)} = \exp\left(n/p_1^{\ell-1}\right) - 1$.

We recognize three regions of interest:

1) $d_t \leq d_u = \sqrt{N/K}$: For small distances we used a dashed line because the failure probability would be zero if we execute the CR decoder of Algorithm 1 componentwise on the ICR codewords.
2) $d_u < d_t \leq d_b$: Within this regime of distances, the failure probability of our decoder stagnates with the same value as in d_b:

$$\mathbb{P}_f \leq U_{\mathbb{P}_f}(d_t) \leq U_{\mathbb{P}_f}(d_b) \leq 2\left(\exp\left(\frac{n}{p_1^{\ell-1}}\right) - 1\right).$$

3) $d_b < d_t < d_{max}$: For this regime of distances the upper bound on the failure probability behaves like $2^{-(\ell+1)(d_{max}-d_t)}$. It increases exponentially and reaches the value 1 approximately when $d_t = d_{max}$.

References

[1] W. Li, V. Sidorenko, and J. S. R. Nielsen, "On decoding interleaved chinese remainder codes," in *2013 IEEE International Symposium on Information Theory*, 2013, pp. 1052–1056.

[2] A. Lenstra, H. Lenstra, and L. László, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, 12 1982.

[3] O. Goldreich, D. Ron, and M. Sudan, "Chinese remaindering with errors," *Information Theory, IEEE Transactions on*, vol. 46, pp. 1330 – 1338, 08 2000.

[4] D. Bleichenbacher, A. Kiayias, and M. Yung, "Decoding of interleaved reed solomon codes over noisy data," in *Lecture Notes in Computer Science*, vol. 2719, 2003, pp. 97–108.

[5] M. Ajtai, "The shortest vector problem in L2 is NP-hard for randomized reductions," in *Proceedings of STOC'98*, 1998, pp. 10–19.

[6] O. Goldreich, D. Ron, and M. Sudan, "Chinese remaindering with errors," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1330–1338, 2000.

[7] E. Guerrini, R. Lebreton, and I. Zappatore, "Polynomial Linear System Solving with Errors by Simultaneous Polynomial Reconstruction of Interleaved Reed-Solomon Codes," in *2019 IEEE International Symposium on Information Theory*, 2019, pp. 1542–1546.

[8] ——, "Polynomial Linear System Solving with Random Errors: New Bounds and Early Termination Technique," in *Proceedings of ISSAC'21*, 2021, pp. 171–178.

[9] E. R. Berlekamp and L. R. Welch, "Error correction of algebraic block codes," U.S. Patent 4,633,470, 1986.

[10] J. v. z. Gathen and J. Gerhard, *Modern Computer Algebra*, 3rd ed. New York, NY, USA: Cambridge University Press, 2013.

[11] J. S. R. Nielsen, "Generalised Multi-sequence Shift-Register synthesis using module minimisation," in *2013 IEEE International Symposium on Information Theory*, Jul. 2013, pp. 882–886, iSSN: 2157-8117.

[12] P. Q. Nguyen and D. Stehlé, "LLL on the Average," in *Algorithmic Number Theory*, 2006, pp. 238–256.

[13] A. Brown, L. Minder, and A. Shokrollahi, "Probabilistic decoding of interleaved rs-codes on the q-ary symmetric channel," in *2004 IEEE International Symposium on Information Theory.*, 02 2004, pp. 326 – 326.

[14] G. Schmidt, V. Sidorenko, and M. Bossert, "Collaborative decoding of interleaved reed-solomon codes and concatenated code designs," *IEEE Transactions on Information Theory*, vol. 55, 11 2006.