Probabilistic Analysis of LLL-based Decoder of Interleaved Chinese Remainder Codes
Matteo Abbondati, Antoine Afflatet, Eleonora Guerrini, Romain Lebreton

To cite this version:

HAL Id: lirmm-04030079
https://hal-lirmm.ccsd.cnrs.fr/lirmm-04030079v2
Submitted on 16 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Open licence - etalab
Probabilistic Analysis of LLL-based Decoder of Interleaved Chinese Remainder Codes

Matteo Abbondati, Antoine Afflatet, Eleonora Guerrini and Romain Lebreton

LIRMM, Université de Montpellier, CNRS
Montpellier, France
{abbondati, guerrini, lebreton}@lirmm.fr
antoine.afflatet@etu-umontpellier.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.

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.

There are interesting complexity reductions between the Noisy Interpolation Problem and the Shortest Vector Problem in lattices, [4] fitting the scenario of the decoding of CR (and Reed Solomon) codes. In the decoding problem, it is classical to define a unique decoding radius below which the correctness of the decoder is guaranteed. In this paper we deal with the problem of decoding beyond this unique radius. There exist two main techniques for that: list decoding algorithms [5] which return the list of all codewords within a certain bound, and interleaving algorithms [1] which return a unique interleaved codeword, at the price of having a non zero decoding failure probability. These two techniques are intrinsically different and in this paper we will focus on the interleaving for CR codes. The most famous example of interleaving 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 [6]. 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 [7]. 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 [8].

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 [9], [10].

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 \( \lfloor k \rfloor_m \) or \( (k \mod m) \).

Definition 2.1 (Chinese Remainder Codes): Let \( n \) be a positive integer, \( p_1 < p_2 < \ldots < p_n \) 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

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

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 \( C \) a code \( CR(P; n, K) \). For the transmission of a codeword \( c \) over
a noisy channel, we write the received word \( r = e + e \), 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 \( E_{R_1, R_2} \) between \( R_1 \) and \( R_2 \) as the subset of indices \( i \) such that \( R_1 \neq R_2 \mod p_i \). The error locator \( \Lambda_{R_1, R_2} := \prod_{i \in E_{R_1, R_2}} p_i \) is the product of the corresponding primes. Its complement is the truth locator \( Y_{R_1, R_2} := N/\Lambda_{R_1, R_2} = \prod_{i \notin E_{R_1, R_2}} p_i \). Finally, the weighted Hamming distance between \( R_1 \) and \( R_2 \) is defined as

\[
d(R_1, R_2) = \log(\Lambda_{R_1, R_2}) = \sum_{i \in E_{R_1, R_2}} \log(p_i).
\]

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 \( d(C) \) of a CR(\( P; n, K \)) code \( C \) satisfies \( d(C) \geq \log(N/K) \).

**Proof:** For any \( C_1, C_2 \in C \) with \( 0 \leq C_1 \leq C_2 < K \), we have that \( C := C_2 - C_1 \) belongs to \( C \) 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 \( d(C_1, C_2) = d(0, C) \geq \log(N/K) \). This inequality holds for any pair of codewords \( C_1, C_2 \), which implies that \( d(C) \geq \log(N/K) \) as stated.

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

\[
d(C, R) \leq d_u := \log(N/K)/2 = \log(\sqrt{N/K}). \tag{1}
\]

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

Since \( d_u < d(C)/2 \), unique decoding is possible ; we restate in Algorithm 1 the unique decoding algorithm described by Goldreich et al. [3], which is reminiscent of the Berlekamp-Welch decoder for Reed-Solomon codes [11].

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

\[
\varphi R = \psi \mod N, \tag{2}
\]

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

\[
S_R = \left\{ (\varphi, \psi) \in \mathbb{Z}^2 \left| \begin{array}{c}
 \varphi R = \psi \mod N \\
 0 \leq \varphi \leq \sqrt{NK}, 0 \leq \psi < \sqrt{NK}
\end{array} \right. \right\}.
\]

Finding a non-zero element in \( S_R \) is a classical computer algebra problem named rational reconstruction (see e.g. [12 Section 5.10]). Using such a rational reconstruction algorithm, we can prove that Algorithm 1 is a decoder of CR codes.

**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 \( d(C, R) \leq \log(\sqrt{N/K}) \) or "decoding failure"

1. Compute \((\varphi, \psi) \in S_R \) by rational reconstruction
2. If \( \varphi \) divides \( \psi \) and \( 0 \leq \psi/\varphi < K \) then
3. \[ \text{return } C := \psi/\varphi \]
4. Else
5. \[ \text{return } "\text{decoding failure}" \]

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

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 \( d(C, R) \leq d_u \). Let us show that the elements of \( S_R \) are unique as rationals, meaning that if \((\varphi, \psi) \) and \((\varphi', \psi') \) \( \in S_R \setminus \{(0,0)\} \) then \( \varphi/\psi = \varphi'/\psi' \). Indeed, combining the equations \( \varphi R = \psi \mod N \) and \( \varphi' R = \psi' \mod N \), we obtain \( \varphi' \psi = \varphi \psi' \mod N \). We know that \( 0 \leq \varphi' \psi < N \) and \( 0 \leq \varphi \psi < N \), so \( \varphi/\psi = \varphi'/\psi' \).

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

Next, we assume that Algorithm 1 returns \( C := \psi/\varphi \) with \((\varphi, \psi) \in S_R \setminus \{(0,0)\} \) and \( 0 \leq \psi/\varphi < K \). This last inequality tells us that \( C \) is a codeword. Moreover, \((\varphi, \psi) \in S_R \) implies that \( \varphi R = \psi \mod N \). So for any \( i \in E_{C,R} \), \( p_i \mod (R - C) \) but \( p_i \not\equiv (R - C) \), since \( r_i \neq c_i \). This implies that \( p_i \not\equiv \varphi \).

Altogether, we get that \( \Lambda C \) divides \( \varphi \). Since \((\varphi, \psi) \in S_R \), we have that \( 0 \leq \varphi \leq \sqrt{NK} \) and we can conclude that \( d(C, R) = \log(\Lambda C, R) \leq \log(\varphi) \leq d_u \).

**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 \( C := CR(P; n, K) \). A homogeneous interleaved Chinese remainder code \( ICR(P; n, K) \) is defined as the set of matrices \( C = (c_{i,j})_{1 \leq i \leq n} \) where \( C_{*,j} := (c_{1,j}, \ldots, c_{n,j}) \in C \) are codewords in the ambient space \( \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_n} \), for any \( j = 1, \ldots, t \).

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 \( R \) as a vector \( R = (R_1, \ldots, R_t) \in (\mathbb{Z}_N)^\ell \).
In particular, we can write \( \mathbf{R} = \mathbf{C} + \mathbf{E} \) for an error \( \mathbf{E} \in (\mathbb{Z}_N)^t \). We will now adapt the distance to ICR codes.

Definition 3.2 (Weighted Hamming distance for ICR codes): Let \( \mathbf{R}, \mathbf{R}' \in (\mathbb{Z}_N)^t \), we define the error support \( \mathcal{E}_{\mathbf{R}, \mathbf{R}'} \) as the union \( \bigcup_{j=1}^t \mathcal{E}_{R_j, R'_j} \) 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 \( R_i \neq R'_i \). The error locator \( \Lambda_{\mathbf{R}, \mathbf{R}'} := \prod_{i \in \mathcal{E}_{\mathbf{R}, \mathbf{R}'}} p_i \) is the product of the corresponding primes. Its complement is the truth locator \( Y_{\mathbf{R}, \mathbf{R}'} := N/\Lambda_{\mathbf{R}, \mathbf{R}'} \). The weighted Hamming distance between \( \mathbf{R} \) and \( \mathbf{R}' \) is defined as \( d(\mathbf{R}, \mathbf{R}') := \log(\Lambda_{\mathbf{R}, \mathbf{R}'}). \)

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

\[
\varphi R_j = \psi_j \mod N \quad \text{for all } 1 \leq j \leq \ell. \tag{3}
\]

Note that \( v_C := (\Lambda, \Lambda C_1, \ldots, \Lambda C_t) \) is a solution where \( \Lambda := \Lambda_{\mathbf{C}, \mathbf{R}}. \)

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 \( \ell \) 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 [13]. 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 \( \mathbb{F}_p[x] \)-module. Intuitively, since for CR codes we have Theorem 2.4, we can transposes 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 \( f_1, \ldots, f_n \) is defined as \( \mathcal{L} := \langle \sum_{1 \leq i \leq n} k_i f_i \rangle \) where \( k_i \in \mathbb{Z} \) and \( f_1, \ldots, f_n \in \mathbb{R}^n \). From Equations (3), there exist \( m_1, \ldots, m_t \in \mathbb{Z} \) such that \((\varphi, \psi_1, \ldots, \psi_t) = (\varphi, \varphi R_1 + m_1 N, \ldots, \varphi R_t + m_t N)\). So the set of solutions of Equations (3) is the lattice \( \mathcal{L} \) generated by the rows of the following matrix

\[
\begin{pmatrix}
1 & R \\
0 & N \cdot \text{Id}_t
\end{pmatrix} \in \mathbb{Z}^{(\ell+1) \times (\ell+1)}. \tag{4}
\]

In particular, \( v_C \in \mathcal{L} \).

A. Decoding using a short vector

As for decoding CR codes, the target solution \( v_C \) has small size, so we redefine \( S_R \) as a subset of solutions of Equations (3) with size constraints

\[
S_R = \left\{ (\varphi, \psi_1, \ldots, \psi_t) \in \mathbb{Z}^{t+1} \mid \begin{array}{l}
\varphi R_i = \psi_i \mod N \\
||\varphi|| \leq 2^r, \quad ||\psi_i|| < 2^r K
\end{array} \right\}
\]

for some distance parameter \( \tau \) that we will soon discuss. For this reason, under a certain constraint (Constraint 1), the decoding of ICR codes is solved with an algorithm that provides the shortest non-zero vector of a lattice. This problem is known to be NP-hard [7], but for interleaving codes parameters the dimension of a lattice can be handle easily and one should not focus on the approximation problem. More explicitly, the decoding algorithm failure mostly depends on the fact that beyond the unique radius decoding more than one codewords could fit the seeked solution. 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 \(||v_s|| \leq \gamma ||\lambda_1(\mathcal{L})||_2\), where \( \lambda_1(\mathcal{L}) \) is the shortest non-zero vector of the lattice and \( \gamma \geq 1 \) is the approximation constant of LLL. It is proved that LLL always outputs a \( \gamma \)-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 \( \gamma \approx 1.022\ell^{(\ell+1)} \) on average [4].

Remark that the output of LLL has constraints on the 2-norm \(||\cdot||_2\). However, a short vector for the 2-norm may not belong to \( S_R \) for a small \( \tau \). In order to make the two notions of size match, we need to introduce a scaling operator \( \sigma_\ell : Q^{\ell+1} \rightarrow Q^{\ell+1} \) such that \( \sigma_\ell((v_0, v_1, \ldots, v_t)) = (0_\ell K, v_0, \ldots, v_t) \). We can then define \( \mathcal{L} := \sigma_\ell(\mathcal{L}) \) and \( v_s := \sigma_\ell^{-1}(v_s) \), where \( v_s := \text{LLL}(\mathcal{L}) \) is a short vector. We can now prove our result, which requires the following condition on \( \mathbf{R} \).

Constraint 1: There exists a codeword \( \mathbf{C} \) such that

\[
\gamma \sqrt{\ell + \delta \Lambda_{C, R}} \leq 2r.
\]

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

Lemma 3.3: Assuming Constraint 1 we have that \( v_s \in S_R \).

Proof: Let \( \tilde{v}_C := \sigma_\ell(v_C) \) belongs to \( \mathcal{L} \). Because LLL computes a \( \gamma \)-approximation of the shortest vector, we have \(||\tilde{v}_s||_2 \leq \gamma ||\lambda_1(\mathcal{L})||_2 \leq \gamma ||v_C||_2 \leq \gamma \sqrt{\ell + \delta \Lambda} \), where \( \lambda_1(\mathcal{L}) \) is the actual shortest non-zero vector of \( \mathcal{L} \). Note that \( \tilde{v}_s \in \mathcal{L} \) implies \( v_s \in \mathcal{L} \). Moreover, \(||v_s||_\infty \leq ||\tilde{v}_s||_2 < \gamma \sqrt{\ell + \delta \Lambda} \leq 2r K \) using Constraint 1. So if we denote \( v_s = (\varphi, \psi_1, \ldots, \psi_t) \), then \(||\varphi|| \leq 2r \) and \(||\psi_i|| < 2^r K \) for \( 1 \leq j \leq \ell \). Altogether, we have proved that \( v_s \in S_R \).

Note that, assuming Constraint 1 we also have that \( v_C = (\Lambda, \Lambda C_1, \ldots, \Lambda C_t) \) belongs to \( S_R \). We have now all the ingredients to present slightly modified version of the ICR codes decoder of [1].

Algorithm 2: Interleaved CR codes decoder

Input: \( \text{ICR}(P; n, K) \), received word \( \mathbf{R} \), parameter \( \tau \)

Output: A codeword \( \mathbf{C} \) s.t. \( d(\mathbf{C}, \mathbf{R}) \leq \tau \) or “decoding failure”

1. Let \( \tilde{\mathcal{L}} \) be the scaled lattice
2. Compute a short vector \( \tilde{v}_s := \text{LLL}(\tilde{\mathcal{L}}) \)
3. Unscale the vector: \( v_s = (\varphi, \psi_1, \ldots, \psi_t) := \sigma_\ell^{-1}(\tilde{v}_s) \)
4. if \( (||\varphi|| \leq 2^r) \) and (\( \varphi \) divides all the \( \psi_j \)) \( j = 1, \ldots, \ell \) and \( (0 \leq \psi_j/\varphi < K \text{ for all } 1 \leq j \leq \ell) \) then
5. return \( (\psi_1/\varphi, \ldots, \psi_t/\varphi) \)
6. else
7. return “decoding failure”
8. 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 $C$ on input $R$ and parameter $\tau$, then $C$ is a codeword of ICR($P; n, K$) such that $d(C, R) \leq \tau$.

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

B. Distribution of random received word

The decoder described in Algorithm 2 must occasionally fail, since it tries uniquely to decode the received word above the unique decoding capacity. In order to analyze our algorithm, we consider a random received word $R$. Similarly to the literature on IRS [6], [15], [16], the distribution $D_{C, \mathcal{E}}$ of $R$ is related to a codeword $C$ and a support for random errors $\mathcal{E}_r \subseteq \{1, \ldots, n\}$. Then, the distribution $D_{C, \mathcal{E}}$ is obtained by taking $R = C + E$ with $E$ as follows: if $i \notin \mathcal{E}_r$, then $E_{i, r} := (e_{i, 1}, \ldots, e_{i, r})$ is the zero vector. If $i \in \mathcal{E}_r$, then the $\ell$ entries of $E_{i, r}$ are mutually independent random variables following the uniform distribution on $Z_{p_i}$, which we will denote as $E_{i, r} \sim \mathcal{U}_\parallel(Z_{p_i})$.

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

Theorem 3.5 (Main result): Given an ICR code $C$ with parameters $N, K, \ell$, and the approximation constant $\gamma$ of LLL, set

$$d_{\text{max}} := \frac{\ell}{\ell + 1} \left(\log(N/K) - \log(6\gamma\sqrt{\ell + 1})\right).$$

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

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

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

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

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 $S_R \subseteq Z_{\Lambda_r}$. When this result is verified, then $v_r$ is colinear to $v_C$ 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 $R$ is drawn at random from the distribution $D_{C, \mathcal{E}}$ with $\log \Lambda_r \leq d_i$. For the analysis, we set $\tau = \tau_i$ in $S_R$, where $\tau_i$ is defined in Theorem 3.5.

A. Decoding failure probability

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

$$S_R \subseteq (\Lambda, \Lambda C_1, \ldots, \Lambda C_\ell)Z.$$

Recall that $\Lambda$ is a short notation for $\Lambda C R$, and also $\mathcal{E} := \mathcal{E}_{C, R}$. To study the condition [6], we need to introduce the set

$$S_{E'} := \{\varphi \in Z_{\Lambda_r} \mid |\varphi E'| \text{ crem } \Lambda_r \leq 2^{n+1}K\Lambda_r/N\}$$

where $e$ crem $m$ denotes the central remainder of $e$ modulo $m$, that is the unique representative of $e \mod m$ in the range $\{-\lceil \frac{m}{2} \rceil + 1, \ldots, \lfloor \frac{m}{2} \rfloor\}$. 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: $2^{n+1}K \leq N$.

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

Proof: Let $(\varphi, \psi_1, \ldots, \psi_\ell) \in S_R$ then

$$\varphi E_i = \varphi (R_i - C_i) = \varphi i - \varphi C_i \mod N.$$  

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

$$|\psi'_i| < \Lambda_r.$$  

Dividing every term in the equation [7] by $Y_r$ we obtain

$$\varphi E'_r = \varphi i \mod \Lambda_r.$$  

Thus, $\varphi \mod \Lambda_r$ belongs to $S_{E'}$, 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 $E'_r = (\varphi E_i) / Y_r$. Therefore, using Equation [9], we get $\psi'_i = 0 mod \Lambda_r$ 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 $(\varphi, \psi_1, \ldots, \psi_\ell) \in (\Lambda, \Lambda C_1, \ldots, \Lambda C_\ell)\mathbb{Z}$.

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 $v_r$ obtained from LLL is not colinear to $v_C$. Since $v_r \in S_R$, this can happen only if $S_R \not\subseteq v_C \mathbb{Z}$. The previous lemma can be restated as

$$S_R \not\subseteq v_C \mathbb{Z} \Rightarrow S_{E'} \not\subseteq \Lambda(\mathbb{Z}_{\Lambda_r}) \Rightarrow S_{E'} \neq \{0\}.$$
Thus, we can say that the decoding failure probability \( P_f \) of Algorithm 2 satisfies \( P_f < P(S_{E'} \neq \{0\}) \). In order to analyze this probability, we introduce the notations \( B := 2^r\sigma_{\Lambda_r} \) and \( g_{\varphi} = \gcd(\varphi, \Lambda_r) \) for \( \varphi \in \mathbb{Z}_{\Lambda_r} \).

**Lemma 4.2:** Given \( E' \sim \mathcal{U}_U((\mathbb{Z}_{\Lambda_r})^\ell) \), then

\[
P(S_{E'} \neq \{0\}) \leq \frac{1}{\Lambda_r} \sum_{\varphi=1}^{\Lambda_r-1} \left( 2 \left| \frac{B}{g_{\varphi}} \right| g_{\varphi} + g_{\varphi} \right)^\ell.
\]

**Proof:** We will denote \( \mathbb{Z}^0_{\Lambda_r} := \mathbb{Z}_{\Lambda_r} \setminus \{0\} \).

\[
P(S_{E'} \neq \{0\}) \leq \mathbb{P} \left( \bigcup_{\varphi \in \mathbb{Z}^0_{\Lambda_r}} \bigcap_{i=1}^{\ell} \left| \varphi E'_i \bmod \Lambda_r \right| \leq B \right)
\]

\[
\leq \sum_{\varphi \in \mathbb{Z}^0_{\Lambda_r}} \mathbb{P} \left( \bigcap_{i=1}^{\ell} \left( \left| \varphi E'_i \bmod \Lambda_r \right| \leq B \right) \right).
\]

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 \( \varphi \in \mathbb{Z}^0_{\Lambda_r} \). So, we can write

\[
P(S_{E'} \neq \{0\}) \leq \sum_{\varphi=1}^{\Lambda_r-1} \left( \mathbb{P}(\varphi U \bmod \Lambda_r \leq B) \right)^\ell
\]

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

\[
O_{\varphi} = \left\{ 0, \varphi, 2\varphi, \ldots, \left( \frac{\Lambda_r}{g_{\varphi}} - 1 \right) \varphi \right\} \subseteq \mathbb{Z}_{\Lambda_r}.
\]

Since for every \( \varphi \in \mathbb{Z}^0_{\Lambda_r} \) 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 \( \mathbb{Z}_{\Lambda_r} \), we have that \( \varphi U \) is also a uniform distribution on \( \mathbb{Z}^0_{\Lambda_r} \). Therefore \( \varphi U \sim \mathcal{U}(O_{\varphi}) \).

\[
\mathbb{P}(\varphi U \bmod \Lambda_r \leq B) = \frac{\# \{ y \in O_{g_{\varphi}} \text{ s.t. } \varphi y \bmod \Lambda_r \leq B \}}{\# \mathbb{O}_{g_{\varphi}}}.
\]

\[
= 2 \cdot \# \left\{ 1 \leq s \leq \frac{\Lambda_r}{g_{\varphi}} - 1 \left| g_{\varphi} s \leq B \right\} + 1 = \frac{g_{\varphi} \left( B \right| g_{\varphi} + 1 \right)}{\Lambda_r}.
\]

The inequality comes from the fact that, within the orbit \( O_{g_{\varphi}} \), we count the remainders \( \bmod \Lambda_r \) which are less than or equal to \( B \) by doubling the number of remainders \( \bmod \Lambda_r \) between 1 and \( B \) and 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:**

\[
P(S_{E'} \neq \{0\}) \leq \left( \frac{2^{\tau+1} K}{N} \right)^\ell \Lambda_r + \exp \left( \frac{n}{p_1^{t-1}} \right) - 1. \tag{10}
\]

**Proof:** Thanks to the previous lemma, we need to obtain an upper bound for the sum \( \frac{1}{\Lambda_r} \sum_{\varphi=1}^{\Lambda_r-1} \left( 2 \right| \frac{B}{g_{\varphi}} g_{\varphi} + g_{\varphi} \right)^\ell \). As the sum is over all the elements \( \varphi \in \mathbb{Z}^0_{\Lambda_r} \) 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 \( \Lambda_r \). For every given divisor \( d < \Lambda_r \) of \( \Lambda_r \), the number of \( \varphi \in \mathbb{Z}^0_{\Lambda_r} \) with \( g_{\varphi} = d \) is equal to \( \phi(\frac{\Lambda_r}{d}) \), where \( \phi \) 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 \( \phi(\frac{\Lambda_r}{d}) \).

Thus we can write

\[
\sum_{\varphi=1}^{\Lambda_r-1} \left( 2 \right| \frac{B}{g_{\varphi}} g_{\varphi} + g_{\varphi} \right)^\ell = \sum_{d|\Lambda_r} \phi(\frac{\Lambda_r}{d}) \left( 2 \right| \frac{B}{d} + d \right)^\ell.
\]

Now we distinguish two cases: if \( d > B \) then \( \left| \frac{B}{d} \right| = 0 \) and the generic term of the sum reduces to \( \phi(\frac{\Lambda_r}{d}) d^\ell \), while when \( d \leq B \) we upper bound the generic term of the sum with \( \phi(\frac{\Lambda_r}{d}) (3B)^\ell \). As 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:

\[
\frac{\Lambda_r}{\Lambda_r^\ell} \sum_{d|\Lambda_r} \phi(\frac{\Lambda_r}{d}) (3B)^\ell + d^\ell
\]

\[
= \left( \frac{3B}{\Lambda_r} \right)^\ell \sum_{d|\Lambda_r} \phi \left( \frac{\Lambda_r}{d} \right) + \frac{1}{\Lambda_r} \sum_{d|\Lambda_r} \phi \left( \frac{\Lambda_r}{d} \right) d^\ell
\]

\[
\leq \left( \frac{3B}{\Lambda_r} \right)^\ell \Lambda_r + \frac{1}{\Lambda_r^\ell - 1} \sum_{d|\Lambda_r} d^\ell - 1
\]

Studying the second sum, we recognize the divisor sum function \( \sigma_{\ell-1}(\Lambda_r) := \sum_{d|\Lambda_r} d^\ell - 1 \). 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

\[
\frac{1}{\Lambda_r^\ell - 1} \sum_{d|\Lambda_r} d^\ell - 1 = \prod_{i \in \mathcal{E}_{\varphi}} \left( 1 + \frac{1}{p_i^{r-1}} \right) = \prod_{i \in \mathcal{E}_{\varphi}} \left( 1 + \frac{1}{p_i^{r-1}} \right) \leq \left( 1 + \frac{1}{p_1^{r-1}} \right)^{|\mathcal{E}_{\varphi}|}
\]

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

\[
\left( 1 + \frac{1}{p_1^{r-1}} \right)^{|\mathcal{E}_{\varphi}|} = \exp \left( |\mathcal{E}_{\varphi}| \ln \left( 1 + \frac{1}{p_1^{r-1}} \right) \right)
\]

\[
\leq \exp \left( |\mathcal{E}_{\varphi}| \frac{1}{p_1^{r-1}} \right) \leq \exp \left( \frac{n}{p_1^{r-1}} \right)
\]

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

**B. Proof of the main Theorem**

**Proof of Theorem 2.5:** We consider the execution of Algorithm 2 with the input parameter \( \tau \) and for a random received word \( R \). Let us first verify that our choice of parameters satisfy Constraint 1 so that \( v_s \in S_R \). Indeed, \( \Lambda \leq \Lambda_r \leq 2^{\tau t} = 2^{\tau (\sqrt{\ell - 1})} \).

[1]
Next, we need to verify Constraint[2] so that we can apply Lemma[3,4] and deduce that the probability of failure of Algorithm[2] is upper-bounded by $P_{E'}(S_{E'} \neq \emptyset)$. We can prove that $2^{\tau_{1}+1} K/N \leq 1$. Note first that

$$2^{\tau_{1}+1} K/N = 2^{(2^{\tau_{1}} \gamma \sqrt{\ell+1}) K/N} \leq 2^{\max(2\gamma \sqrt{\ell+1} K/N)} = \left( \frac{6 \gamma \sqrt{\ell+1} K/N}{\ell^{1/(\ell+1)}} \right)^{2} \gamma \sqrt{\ell+1} K/N$$

$$= \left( \frac{2 K \gamma \sqrt{\ell+1}}{N} \right)^{1/(\ell+1)}.$$

We can assume $2 \leq K \leq N$ since we took $k < n$ after Definition[2]. Moreover, the approximation constant $\gamma$ of LLL always satisfies $\gamma \leq \sqrt{2}$. 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 $\ell \in N$. Altogether, we have proved $2^{\tau_{1}+1} K/N \leq 1$ and Constraint[2].

At this point, we can apply Lemma[4,5] to get an upper-bound on the failure probability. It remains to prove that

$$\left( \frac{2^{\tau_{1}+1} K}{N} \right)^{\ell} \Lambda_{r} \leq 2^{2^{d_{\text{max}}+1} - d_{1}}.$$

Since $2^{2^{d_{\text{max}}}(\ell+1)} = (6 K \gamma \sqrt{\ell+1} K/N)^{\ell}$, the previous inequality is equivalent to $\Lambda_{r} 2^{\gamma \ell \sqrt{\ell+1}} \leq 2^{d_{1}(\ell+1)} (\gamma \sqrt{\ell+1})^{\ell}$, which is verified since $2^{\gamma \ell} = 2^{d_{1}} \gamma \sqrt{\ell+1}$ and $\Lambda_{r} \leq 2^{d_{1}}$. We recognize three regions of interest:

1. $d_{1} \leq d_{u} = \sqrt{N/K}$: 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. $d_{u} < d_{1} \leq d_{b}$: Within this regime of distances, the failure probability of our decoder stagnates with the same value as in $d_{b}$:

$$P_{f} \leq U_{p_{1}}(d_{1}) \leq U_{p_{1}}(d_{b}) \leq 2 \left( \exp \left( \frac{n}{p_{1}^{\ell-1}} \right) - 1 \right).$$

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

If we were to use an exact SVP solver, this would correspond to having $\gamma = 1$ and the only difference would be that both $d_{b}$ and $d_{\text{max}}$ increase of the same quantity $r^{1/\ell} \log(\gamma).$

**REFERENCES**


