21st International Symposium on
Mathematical Theory of Networks and Systems
July 7-11, 2014. Groningen, The Netherlands

On Decoding of Interleaved Chinese Remainder Codes

Wenhui Li, Johan S. R. Nielsen and Vladimir Sidorenko

***Abstract*— The decoding of the Interleaved Chinese Remainder (ICR) code is reformulated as finding a short vector in a $\mathbb{Z}$-module and a decoding algorithm for ICR codes is proposed. The complexity of our algorithm is linear in the length of the code. We give a proof of the failure probability and derive the decoding radius.**

***Index Terms*— Interleaved Chinese Remainder codes, Lattice reduction**

I. Introduction

The Chinese remainder (CR) code can be efficiently applied for distributive computations and for secret sharing [1]. Several decoding algorithms [1], [2], [3] has been studies to decode a single CR code. In this paper, we consider interleaved Chinese remainder (ICR) codes. The ICR codes are similar to interleaved Reed–Solomon (IRS) codes in many aspects. Recently, the construction of IRS codes have been intensively studied in several publications, e.g. [4] where decoding beyond half the minimum distance is allowed. Among these decoding algorithms for IRS codes, Nielsen [5] proposed a module minimization approach for solving multiple Key Equations by finding short vectors in a certain space.

In this paper we adapt this approach for decoding ICR codes. In Section II, after the CR and ICR codes are introduced, we propose a decoding algorithm for ICR codes. In comparison with [6], we focus on proving the failure probability of the algorithm and give a decoding radius afterwards. These are shown in Section II-A and II-B. Complexity and test results are shown in the end.

II. Decoding Interleaved Chinese Remainder Code

Interleaving is a scheme which is often used in a bursty channel. The codeword is a matrix where each row form a codeword from some component code. The interleaved Reed–Solomon(IRS) code has been studied, e.g, [4]. In [6], this scheme was applied to the CR code, and a decoding algorithm was proposed for ICR codes. However, the failure probabilities were not completely shown and the decoding radius was analyzed by our intuitive guess. In this work, we continue our study of proving the failure probability and propose the decoding radius according to the failure probability.

The work of W. Li and V. Sidorenko is supported by the German Research Council (Deutsche Forschungsgemeinschaft DFG) under project Bo 867/22.

W. Li, J. S. R. Nielsen and V. Sidorenko are with Institute of Communications Engineering, University of Ulm, Ulm, Germany wenhui.li,vladimir.sidorenko@uni-ulm.de and jsrn@jsrn.dk

Now let us introduce the Chinese remainder code. Let n be the code length and $0 < p_1 < p_2 < \cdots < p_n$ a list $\mathcal{P}$ of n relatively prime positive integers. We construct a polyalphabetic code, where the i-th component of codeword $\mathbf{c} = (c_1, c_2, \ldots, c_n)$ is taken from the alphabet $\mathbb{Z}_{p_i}$, being the ring of integers modulo p_i. Thus the codewords are selected from the code space $\mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2} \times \cdots \times \mathbb{Z}_{p_n}$ of size $N = p_1 p_2 \ldots p_n$. We also need a *cardinality* K which is defined by $K = \prod_{i=1}^{k} p_i, 0 < k \leq n$. We denote the remainder when $x \in \mathbb{Z}$ is divided by $y \in \mathbb{Z}$, by $[x]_y, 0 \leq [x]_y \leq y - 1$.

Definition 1 (Chinese Remainder Code)**.** *A Chinese Remainder code $\mathcal{CR}(\mathcal{P}; n, K)$ having cardinality K and length n with alphabet sizes $\mathcal{P}$ is defined as follows*

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

A CR code is *Maximum Distance Separable* (MDS) [1]. Assume we transmit some codeword $\mathbf{c} = (c_1, \ldots, c_n) \in \mathcal{CR}(\mathcal{P}; n, K)$ over an additive noisy channel and receive the word $\mathbf{r} = \mathbf{c} + \mathbf{e}$ where $\mathbf{e} = (e_1, \ldots, e_n), e_i \in \mathbb{Z}$, is the error vector. Letting $\mathbf{r} = (r_1, \ldots, r_n)$ we have $r_i = [c_i + e_i]_{p_i}, i = 1, \ldots, n$. Using the Chinese remainder theorem (CRT), we compute R such that $[R]_{p_i} = r_i$, and likewise an E such that $[E]_{p_i} = e_i$; we then know $R \equiv C + E \mod N$. Since $[E]_{p_i} = 0$ for all i such that $r_i = c_i$, we know E has factors of all the primes from the non erroneous positions.

The receiver can reconstruct C from any k correct positions in $\mathbf{r}$ by the Chinese remainder theorem; a common decoding strategy we use in this paper, is therefore first to identify the erroneous positions. The positions of the errors can be found by determining factors of the *error-locator* Λ, defined as $\Lambda = \prod_{i:r_i \neq c_i} p_i$.

Note that the logarithm of Λ is the weighted Hamming distance between $\mathbf{r}$ and $\mathbf{c}$, and $\Lambda E \equiv 0 \mod N$. Adding ΛC on both side immediately leads to a Key Equation:

$$\Lambda R \equiv \Lambda C \mod N. \tag{1}$$

In [1], a decoding algorithm is proposed which finds Λ in (1) if

$$\Lambda \leq \sqrt{N/(K-1)}). \tag{2}$$

Let t be the number of errors, using $\Lambda < {p_n}^t$, we get a decoding radius in the Hamming metric

$$t \leq \left\lfloor \frac{1}{2} \cdot \frac{\log(N/K)}{\log p_n} \right\rfloor. \tag{3}$$

Arranging ℓ CR codewords row wisely as a $\ell \times n$ matrix, we have the interleaved Chinese remainder (ICR) code.

ISBN: 978-90-367-6321-9

Definition 2 (Interleaved Chinese Remainder Code)**.** *Consider ℓ CR codes $\mathcal{CR}(\mathcal{P}; n, K_l), l \in 1, \ldots, \ell$. Denote the list $K_1, K_2, \ldots, K_\ell$ by $\mathcal{K}$. The Interleaved Chinese Remainder code $\mathcal{ICR}(\mathcal{P}; n, \mathcal{K})$ is defined as the set of matrices*

$$\begin{pmatrix} c_1^{(1)} & c_2^{(1)} & \ldots & c_n^{(1)} \\ c_1^{(2)} & c_2^{(2)} & \ldots & c_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ c_1^{(\ell)} & c_2^{(\ell)} & \ldots & c_n^{(\ell)} \end{pmatrix}$$

where $\mathbf{c}^{(l)} = (c_1^{(l)}, \ldots, c_n^{(l)}) \in \mathcal{CR}(n, K_l), l = 1, \ldots, \ell$.

For the remainder of this section, consider some received matrix with rows $\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(\ell)}$ each with $\mathbf{r}^{(l)} = \mathbf{c}^{(l)} + \mathbf{e}^{(l)}$ for some error row $\mathbf{e}^{(l)}$. We now define a complete error-locator which identifies all columns having any errors, i.e,

$$\Lambda = \prod_{i:\exists l: r_i^{(l)} \neq c_i^{(l)}} p_i.$$

When we refer to "the number of errors", it is also the number of factors in the above product.

For each $\mathbf{c}^{(l)}$ and $\mathbf{r}^{(l)}$ corresponds a $C^{(l)}$ respectively $R^{(l)}$. For a particular row l, even though the complete error-locator Λ might be a multiple of that row's error-locator, the Key Equation (1) still holds with Λ; thus, to collaboratively decode the ICR code, we want to solve a system of ℓ Key Equations as follows

$$\begin{cases} \Lambda R^{(1)} & \equiv & \Lambda C^{(1)} & \mod N \\ \Lambda R^{(2)} & \equiv & \Lambda C^{(2)} & \mod N \\ \quad \vdots & & & \\ \Lambda R^{(\ell)} & \equiv & \Lambda C^{(\ell)} & \mod N \end{cases}. \tag{4}$$

Recently, Nielsen [5] used a module minimization approach to solve multiple Key Equations over some polynomial ring $\mathbb{F}[x]$, such as those arising when decoding Interleaved Reed–Solomon codes. We will apply essentially the same approach for our Key Equations, but the algebraic differences between $\mathbb{F}[x]$ and $\mathbb{Z}$ implies fundamental differences in the final algorithms.

The l-th Key Equation means that there exists some $v_l \in \mathbb{Z}$ such that $\Lambda R^{(l)} - v_l N = \Lambda C^{(l)}$. We can collect these ℓ equations into one in a vectorized form and say that $\mathbf{s} = (\Lambda, \Lambda C^{(1)}, \ldots, \Lambda C^{(\ell)})$ must be a vector in the $\mathbb{Z}$-row space of the matrix

$$\mathbf{M} = \begin{pmatrix} 1 & R^{(1)} & R^{(2)} & \ldots & R^{(\ell)} \\ 0 & N & 0 & \ldots & 0 \\ 0 & 0 & N & \ldots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \ldots & N \end{pmatrix}. \tag{5}$$

The crucial observation is now that whenever few errors have occurred, $\mathbf{s}$ is often the *shortest* vector in the row space of $\mathbf{M}$; we will explain and formalize this later with Theorem 1. By "short" we mean the L_2 norm, and to increase the probability that $\mathbf{s}$ is the shortest vector, we will actually regard the row space of $\mathbf{M}_\omega$, a weighted version of $\mathbf{M}$, where we scale the i-th column with some $\omega_i \in \mathbb{Z}$. We are thus seeking a $\mathbf{s}_\omega = (\Lambda\omega_0, \Lambda C^{(l)}\omega_1, \ldots, \Lambda C^{(l)}\omega_\ell)$. We get back to how exactly we assign the ω_i in Corollary 1.

We reformulate solving Key Equations (4) to finding a shortest vector in the $\mathbb{Z}$-module. Computing the shortest vector in the row space of a matrix under the L_2 norm is unfortunately an $\mathcal{NP}$-hard problem [7]; however, the Lenstra–Lenstra–Lovász (LLL) algorithm [8] is an efficient method to find a vector which is *close* to the shortest one, i.e. it finds a vector whose L_2 norm is at most $\gamma\|\mathbf{v}\|$, where $\mathbf{v}$ is a shortest vector and γ is a constant. In the worst case, $\gamma = \sqrt{2}^{\ell+1}$, where $\ell + 1$ is the dimension of the row space; however, experiments indicate that in random instances the LLL and its modifications usually do much better, with $\gamma \approx 1.02^{\ell+1}$ [9]. To be certain that our computation will lead us to $\mathbf{s}_\omega$, we must therefore not only be sure that $\mathbf{s}_\omega$ is the shortest vector in the row space, but that there are no other vectors of length at most $\gamma\|\mathbf{s}_\omega\|$.

Theorem 1 essentially says that whenever not too many errors have occurred, this is indeed almost always the case. Therefore, one can construct $\mathbf{M}_\omega$, apply the LLL algorithm to find a short vector in it, and with high probability, the output will be $\mathbf{s}_\omega$. This immediately leads to the decoding algorithm given as Algorithm 1.

Algorithm 1: Decoding ICR code $\mathcal{ICR}(n, \mathcal{K})$

Input: The lists $\mathcal{P}$ and $\mathcal{K}$, the received words $\mathbf{r}^{(l)},\ l = 1, \ldots, \ell,\ N$

Output: The error-locator Λ or Fail

Preprocessing: $\omega_0, \ldots, \omega_\ell$ according to Corollary 1

1. Compute $R^{(1)}$, …, $R^{(\ell)}$ using CRT.
2. Construct $\mathbf{M}$ as in (5) and multiply the ith column by ω_i for $i = 0, \ldots, \ell$.
3. Run the LLL algorithm which returns a short vector $\mathbf{v}_\omega$.
4. If the zeroth position of $\mathbf{v}_\omega$ has the form $\omega_0\Lambda$ where Λ is a valid error-locator, then return Λ. Otherwise, return Fail.

A. Failure Probability

With the overall idea explained, we can go on to analyze the probability that the above algorithm will fail, and from this derive how to assign the ω_i. Our failure probability will depend on the unknown Λ, but we discuss in Section II-B how this can be interpreted as a decoding radius.

The algorithm fails when there is a vector in the row space $\mathbf{M}_\omega$ different from $\mathbf{s}_\omega$ but which has L_2 norm within $\gamma\|\mathbf{s}_\omega\|$, and we will upper bound this probability. Our theorem will assume that certain values behave as independent, uniformly distributed random variables, and so we will need the following lemma:

Lemma 1. *Given some $N, T \in \mathbb{Z}$ with $T < N$ and $c_1, \ldots, c_\ell \in \mathbb{Z}_+$, and let $X_1, \ldots, X_\ell$ be independent random variables, uniformly distributed on $[0, N-1]$. Then*

$$P = \text{Prob}[c_1 X_1 + \ldots + c_\ell X_\ell < T] \leq \frac{T^\ell}{\ell! N^\ell c_1 \cdots c_\ell}. \tag{6}$$

Proof. To prove the lemma we relax integer variables X_i to real valued variables uniformly distributed on $[0, N]$ and show that for this case (6) holds with equality. If N is large, which will be our case, this relaxation gives a very precise upper bound (6) for the probability P in the lemma.

Assume that X_i are real valued variables uniformly distributed on $[0, N]$ with density $p[X_i = b] = 1/N$ for $b \in [0, N]$ and define the sum $S_i = c_1X_1 + \ldots + c_iX_i$ for $1 < i \leq \ell$. Then we have

$$\begin{aligned} P[S_i < T] &= \int_0^{T/c_i} p[X_i = b]P[S_{i-1} < T - c_ib_i]\ db \\ &= \frac{1}{N}\int_0^{T/c_i} P[S_{i-1} < T - c_ib_i]\ db. \end{aligned} \tag{7}$$

We proceed by induction on i. For the base case $i = 1$, observe that

$$P[S_1 < T] = P[c_1X_1 < T] = P[X_1 < T/c_1] = \frac{T}{Nc_1}.$$

For the induction step, we continue from (7) using the induction hypothesis:

$$\begin{aligned} P[S_i < T] &= \frac{1}{N}\int_0^{T/c_i} P[S_{i-1} < T - c_ib]\ db \\ &= \frac{1}{N}\int_0^{T/c_i} \frac{(T - c_ib)^{i-1}}{(i-1)!N^{i-1}c_1 \ldots c_{i-1}}\ db \\ &= \frac{c_i^{i-1}}{(i-1)!N^ic_1 \ldots c_{i-1}}\int_0^{T/c_i} \left(\frac{T}{c_i} - b\right)^{i-1} db. \end{aligned}$$

Since

$$\int_0^{T/c_i} \left(\frac{T}{c_i} - b\right)^{i-1} db = \frac{1}{i}\left(\frac{T}{c_i}\right)^i$$

we have proved the step of induction and for $i = \ell$ we obtain the expression of the lemma. □

Theorem 1. *Let A be a random variable, uniformly distributed on $1, \ldots, \lfloor T/\omega_0 \rfloor$, where T is defined below, and assume then that we can regard $(AR^{(l)} \mod N)$ for $l = 1, \ldots, \ell$ as ℓ independent random variables, uniformly distributed on $0, \ldots, N-1$.*

Assume that the LLL algorithm finds a vector whose L_2 norm is at most $\gamma\|\mathbf{v}_\omega\|$ where $\mathbf{v}_\omega$ is a shortest vector in the row space of $\mathbf{M}_\omega$. For a random error-locator Λ, the probability of decoding failure $P_f(\Lambda)$ satisfies

$$P_f(\Lambda) \leq 1 - \left(1 - \frac{T^\ell}{\ell!N^\ell\omega_1 \cdots \omega_\ell}\right)^{T/\omega_0}$$

where $T = \tilde{\gamma}\max\{\omega_0\Lambda, \omega_1\Lambda K_1, \ldots, \omega_\ell\Lambda K_\ell\}$ and $\tilde{\gamma} = \sqrt{\gamma(\ell+1)}$.

Proof. We consider two cases.

Case 1: If $p_i \nmid R^{(l)}, \forall i = 1, \ldots, n, l = 1, \ldots, \ell$, then the variables $(AR^{(l)} \mod N)$ are uniformly distributed in $[0, \ldots, N-1]$.
The decoder can only fail if there is a vector $\mathbf{v}_\omega \neq \mathbf{s}_\omega$ with $\|\mathbf{v}_\omega\| < \gamma\|\mathbf{s}_\omega\|$, i.e,

$$\sum_{l=0}^{\ell}(\omega_lv_l)^2 < \gamma\left((\omega_0\Lambda)^2 + \sum_{j=1}^{\ell}(\omega_j\Lambda C^{(j)})^2\right)$$

where ω_lv_l are the components of $\mathbf{v}_\omega$. Let $\tilde{T} = \max\{\omega_0\Lambda, \omega_1\Lambda K_1, \ldots, \omega_\ell\Lambda K_\ell\}$; then the above can only occur if $\sum_{l=0}^{\ell}(\omega_lv_l)^2 < \gamma(\ell+1)\tilde{T}^2$. Due to the Cauchy–Schwartz inequality, that implies

$$\sum_{l=0}^{\ell}\omega_lv_l < \sqrt{\gamma(\ell+1)}\tilde{T} = T. \tag{8}$$

We will upper bound $P_f(\Lambda)$ by the probability that a vector $\mathbf{v}_\omega \neq \mathbf{s}_\omega$ satisfying (8) is in the row space. Such a vector can be written in the form

$$\big(\omega_0A,\ \omega_1(AR^{(1)} \bmod N),\ \ldots,\ \omega_\ell(AR^{(\ell)} \bmod N)\big)$$

where $\omega_0A \in \{1, \ldots, T-1\}$. Now we use Lemma 1 to over-approximate the probability that for a given $A \in \{1, \ldots, \lfloor T/\omega_0 \rfloor\}$, the associated vector of the above form satisfies (8):

$$\begin{aligned} P &= \text{Prob}\Big[\ \sum_{j=1}^{\ell}\omega_i(AR^{(j)} \bmod N) < T - \omega_0A\ \Big] \\ &< \text{Prob}\Big[\ \sum_{j=1}^{\ell}\omega_i(AR^{(j)} \bmod N) < T\ \Big] \\ &\leq \frac{T^\ell}{\ell!N^\ell\omega_1 \ldots \omega_\ell}. \end{aligned} \tag{9}$$

Thus the probability that none of the T/ω_0 choices of A satisfies (8) becomes at least $(1-P)^{T/\omega_0}$, and the statement follows.

Case 2: If for some $i \in [1, n]$ and some $l \in [1, \ell]$, $p_i|R^{(l)}$, then $p_i|(AR^{(l)} \mod N)$. Therefore, the variables $(AR^{(l)} \mod N)$ are clearly not uniformly distributed in $\{0, 1, \ldots, N-1\}$, but still behave close to uniformly distributed in $\{0, p_i, 2p_i, \ldots, N - p_i\}$. In this case, the vector $\mathbf{v}_\omega$ can be written in the following way:

$$\big(\omega_0A,\ \omega_1([AR^{(1)}]_N), \ldots, \omega_ip_i[\frac{AR^{(l)}}{p_i}]_{\frac{N}{p_i}}, \ldots, \omega_\ell([AR^{(\ell)}]_N)\big).$$

Assume there are other p_j which divide any $R^{(l)}$. We proceed as in Case 1. In the end we reach the same failure probability, since

$$\frac{T^\ell}{\ell!N^{\ell-1}\frac{N}{p_i}\omega_1 \ldots \omega_ip_i \ldots \omega_\ell} = \frac{T^\ell}{\ell!N^\ell\omega_1 \ldots \omega_\ell}. \tag{10}$$

If there are other p_j which divide other $R^{(l)}$, same arguments can be iterated. □

Though we have not proved that the following choice of weights is optimal, it seems intuitive:

Corollary 1. *With the weights chosen as $\omega_0 = K_\ell$ and $\omega_i = K_\ell/K_i,\ i = 1, \ldots, \ell$, the failure probability becomes*

$$P_f(\Lambda) \leq 1 - \left(1 - \frac{\tilde{\gamma}^\ell\Lambda^\ell\prod_{l=1}^{\ell}K_l}{\ell!N^\ell}\right)^{\tilde{\gamma}\Lambda}. \tag{11}$$

B. Decoding Radius

To decode a single CR code, it follows from (2) that the decoding will never fail if $\Lambda \leq \sqrt{N/(K-1)}$). For ICR code, when $\Lambda > \sqrt{N/(K-1)}$, we can also correct errors, but we can not guarantee that our decoding algorithm will always work in this case. Therefore, we define a threshold ϕ, and say that "If Λ is within some range, Algorithm 1 can decode with probability $1-\phi$". If we set the failure threshold as ϕ in (11), i.e,

$$1 - \left(1 - \frac{\tilde{\gamma}^\ell \Lambda^\ell \prod_{l=1}^{\ell} K_l}{\ell! N^\ell}\right)^{\tilde{\gamma}\Lambda} \leq \phi$$

and the power part is approximated as $1 - \frac{\tilde{\gamma}^\ell \Lambda^\ell \prod_{l=1}^{\ell} K_l}{\ell! N^\ell}\tilde{\gamma}\Lambda$ according to the first order truncation of the Binomial series, the error-locator is bounded as

$$\Lambda \leq \left(\frac{\phi \ell!}{[\gamma(\ell+1)]^{\frac{\ell+1}{2}}}\right)^{\frac{1}{\ell+1}} \cdot \left(\frac{N}{\bar{K}}\right)^{\frac{\ell}{\ell+1}} \triangleq \alpha \left(\frac{N}{\bar{K}}\right)^{\frac{\ell}{\ell+1}}, \quad (12)$$

where $\bar{K} = \sqrt[\ell]{K_1 \cdots K_\ell}$ and α is some constant close to 1 which depends on ϕ, γ and ℓ. The decoding radius in Hamming metric of our algorithm, in the above sense, is

$$t \leq \left\lfloor \frac{\ell}{\ell+1} \frac{\log(N/\bar{K})}{\log p_n} + \frac{\log \alpha}{\log p_n} \right\rfloor \quad (13)$$

where the latter term is a negative constant. To decode a single CR code ($\ell = 1, \phi = 1, \gamma = (\sqrt{2})^{\ell+1} = 2$), (13) coincides with (3).

C. Complexity

The complexity of Step 3 in Algorithm 1 determines the overall complexity. By [10, Theorem 16.11], Step 3 performs $O(\ell^4 \log Z)$ operations on integers of bit-length $O(\ell \log Z)$ where Z is the greatest norm of rows in $\mathbf{M}_\omega$. Choosing the ω_i as in Corollary 1, we have $Z = \sqrt{\ell+1} N K_\ell / K_1$ which means $\log Z < 2n\sqrt{\ell+1}\log(p_n)$. Therefore, the complexity of Algorithm 1 is $O(n\ell^{4.5}\log(p_n))$ operations on integers of bit-length $O(n\ell^{1.5}\log(p_n))$.

D. Test Results

As an example, let us consider the ICR code $\mathcal{ICR}(n = 100, \mathcal{K} = [81, 81, 82, 82, 83])$, i.e., interleaving factor $\ell = 5$, and with the prime list $\mathcal{P} = [101, 103, \ldots, 691]$. One can decode up to 14 errors according to (13). For a CR code $\mathcal{CR}(n = 100, k = 81)$ and $\mathcal{CR}(n = 100, k = 83)$, one can decode up to 9 and 8 errors respectively. Table I shows the value of $\log \alpha / \log P_n$ which depends on $\phi, \gamma (= \hat{\gamma}^{\ell+1})$ when $\ell = 5$. We can see that $\log \alpha / \log P_n$ is much smaller than 1 such that one can use $\left\lfloor \frac{\ell}{\ell+1} \frac{\log(N/\bar{K})}{\log p_n} \right\rfloor$ to bound the decoding radius.

Running 10,000 tests with patterns of weights ranging from 14 to 18, we got the test results as given on Table II. For each number of errors t, we then calculated the following aggregate statistics

$$A_{Obs} = \#\text{failures}/\#\text{Tests}_t$$
$$A_T^{\hat{\gamma}} = \textstyle\sum_{\Lambda \in \text{Tests}_t} P_f^{\gamma=\hat{\gamma}^{\ell+1}}(\Lambda)/\#\text{Tests}_t,$$

	$\hat{\gamma}=1$	$\hat{\gamma}=1.02$	$\hat{\gamma}=\sqrt{2}$
$\phi=1$	-0.015	-0.024	-0.174
$\phi=0.9$	-0.018	-0.027	-0.177

TABLE I
THE VALUE OF $\log \alpha / \log p_n$ IN (13) FOR $\ell = 5$

the latter calculated for $\hat{\gamma} \in \{1, 1.02, \sqrt{2}\}$. We have also calculated $P_T^{\hat{\gamma}} = P_f^{\gamma=\hat{\gamma}^{\ell+1}}(\Lambda_{\max})$, i.e., the failure probability of the largest possible Λ.

We see that the upper bounds of $P_T^{\hat{\gamma}}$ are quite pessimistic estimates on the average failure probability, and that it is slightly too optimistic to assume $\hat{\gamma} \leq 1.02$. According to (13) with $\phi = 1$, the decoding radius is $t = 14$, which is also pessimistic.

t	A_{Obs}	A_T^1	$A_T^{1.02}$	$A_T^{\sqrt{2}}$	P_T^1	$P_T^{1.02}$	$P_T^{\sqrt{2}}$
14	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0
16	4.68	3.51	3.81	10.71	99.77	99.98	100
17	89.66	87.25	87.79	94.84	100	100	100
18	99.94	99.95	99.95	100	100	100	100

TABLE II
FAILURE PROBABILITIES % ($n = 100$)

III. CONCLUSION

We propose a decoding algorithm for the ICR codes which has linear complexity in the length of the codeword. We then give a proof of the failure probability. The decoding radius is also analyzed.

REFERENCES

[1] O. Goldreich, D. Ron, and M. Sudan, "Chinese remaindering with errors," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1330–1338, 2000.
[2] V. Guruswami, A. Sahai, and M. Sudan, ""Soft-decision" decoding of Chinese remainder codes," in *Proc. of the 41st IEEE Symposium on Foundations of Computer Science*, 2000, pp. 159–168.
[3] W. Li, "On Syndrome Decoding of Chinese Remainder Codes," in *The 13th Int. Workshop on Algebraic and Combinatorial Coding Theory*, 2012.
[4] G. Schmidt, V. Sidorenko, and M. Bossert, "Collaborative decoding of interleaved Reed–Solomon codes and concatenated code designs," *IEEE Trans. Inform. Theory*, vol. 55, no. 7, pp. 2991–3012, 2009.
[5] J. Nielsen, "Generalised multi-sequence shift-register synthesis using module minimisation," in *IEEE Int. Symposium on Inform. Theory*, 2013, pp. 882–886.
[6] W. Li, V. Sidorenko, and J. S. R. Nielsen, "On decoding Interleaved Chinese Remainder codes," in *IEEE Int. Symposium on Inform. Theory*, 2013, pp. 1052–1056.
[7] M. Ajtai, "The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract)," in *Proc. of the 30th annual ACM symposium on Theory of computing*, 1998, pp. 10–19.
[8] A. K. Lenstra, "Factoring multivariate polynomials over finite fields," *J. Computer System Sciences*, vol. 30, no. 2, pp. 235–248, 1985.
[9] F. Fontein, M. Schneider, and U. Wagner, "A Polynomial Time Version of LLL With Deep Insertions," in *Int. Workshop on Coding and Cryptography*, 2013.
[10] J. v. z. Gathen and J. Gerhard, *Modern Computer Algebra*. Cambridge University Press, Jul. 2003.