# Chinese Remainder Codes: Using Lattices to Decode Error Correcting Codes Based on Chinese Remaindering Theorem

Ragesh Jaiswal
University of California San Diego
(rjaiswal@cs.ucsd.edu)

## 1 Introduction

This report is an incomplete survey of Chinese Remaindering Codes. We study the work of Goldreich, Ron and Sudan [GRS00] and Boneh [B02] which give unique and list-decoding algorithms for an error correcting code based on the Chinese Remaindering Theorem. More specifically, we will look at a decoding algorithm from [GSM00] which uniquely decodes upto $(n-k)\frac{\log p_1}{\log p_1 + \log p_n}$ errors. We will also look at a list-decoding algorithm ([GRS00]) which decodes upto $n - \sqrt{2kn\frac{\log p_n}{\log p_1}}$ error and an improvement ([B02]) to $n - \sqrt{kn\frac{\log p_n}{\log p_1}}$ errors. Here $k$ is the message length, $n$ is the length of the codeword and $p_1$ and $p_n$ are the first and last relatively prime integers used in the CRT code. We will define them in the next section.

**Organization of this report** In the next section we set up the basics required for the rest of the report. In Section 3 we look at an algorithm due to Goldreich, Ron and Sudan [GRS00] for unique decoding. In Section 4 we look at a list-decoding algorithm due to Goldreich, Ron and Sudan [GRS00] and an improvement due to Boneh [B02]. Finally, we point out possible extensions and briefly mention results from [GSM00] which indeed achieves some of the goals perceived.

## 2 Preliminaries

In this section we set up the basics required for the report.

**Definition 1.** Let $p_1 < p_2 < \ldots < p_n$ denote positive integers which are *relatively prime*. We denote $K = p_1 \cdot p_2 \ldots p_k$ (for some $k \leqslant n$) and $N = p_1 \cdot p_2 \ldots p_n$. Furthermore for a subset $A \subseteq [n]$, we denote $P_A = \prod_{i \in A} p_i$.

We will need the following version of the Chinese Remaindering Theorem (CRT) throughout the report. We omit the proof here.

**Theorem 2** (Chinese Remaindering Theorem (CRT))**.** *Given $K = p_1 \cdot p_2 \ldots p_k$ such that $p_i$'s are relatively prime, there is a one-one onto mapping between any integer $x \in \mathbb{Z}_K$ and $\Phi(x) = (x \mod p_1, x \mod p_2, \ldots, x \mod p_k) \in \prod \mathbb{Z}_{p_i}$. In addition, there is an efficient procedure to compute the mapping in either direction.*

The following error correcting code can be designed based on the Chinese Remaindering Theorem above.

**CRT Code** Given relatively prime integers $p_1 < p_i < \ldots < p_n$, $K = \prod_{i=1}^{k} p_i$ and a message $m \in \mathbb{Z}_K$, the CRT code for $m$ is:

$$\bar{m} = CRT(m) =< m_1, \ldots, m_n >, \ where\ m_i = m\ (mod\ p_i)$$

We will need the following definitions for further analysis.

**Definition 3.** Given $\bar{r} =< r_1, \ldots, r_n >, r_i \in \mathbb{Z}_{p_i}$ and $\bar{m} =< m_1, \ldots, m_n >, m_i \in \mathbb{Z}_{p_i}$, consider the sets

$$A(\bar{r}, \bar{m}) = \{i : r_i = m_i\} \quad and \quad D(\bar{r}, \bar{m}) = \{i : r_i \neq m_i\}$$

In simpler words $A$ denote the subset of indices where $\bar{r}$ and $\bar{m}$ *agree* and $D$ denote the subset of indices where they *disagree*. Also, $P_A$ is called the *amplitude* of agreement between $\bar{r}$ and $\bar{m}$ and similarly $P_D$ is called the amplitude of disagreement.

Next, we show some simple properties of the above CRT code. Consider the following theorem which gives a unique decoding property of of the CRT code.

**Theorem 4.** *Given $\bar{r} =< r_1, \ldots, r_n >, r_i \in \mathbb{Z}_{p_i}$. There is a unique message $m \in \mathbb{Z}_N$ such that*

$$|A(\bar{r}, CRT(m))| \geqslant \frac{n+k}{2}$$

*Proof.* For the sake of contradiction, suppose there are two messages $m_1, m_2 \in \mathbb{Z}_K$ such that $|D(\bar{r}, CRT(m_1))| \leqslant (n-k)/2$ and $|D(\bar{r}, CRT(m_2))| \leqslant (n-k)/2$. Let $D_1 = D(\bar{r}, CRT(m_1))$ and $D_2 = D(\bar{r}, CRT(m_2))$. Consider the subset $D = D_1 \cup D_2$ and $A = [n] - D$. Clearly, $|A| \geqslant k$. From CRT we have

$$m_1 \equiv m_2\ (mod\ P_A)$$

but since $K \leqslant P_A$ we get that $m_1 = m_2$. □

The above theorem shows that, information theoretically, it is possible to uniquely decode the CRT code from upto $(n-k)/2$ errors. So, the natural question that arises is, "how much error can be tolerated algorithmically?", or in other words, "what is the bound on the number of error which can be corrected using an efficient algorithm?" We know that by using list-decoding we can go beyond the information theoretic bound. So, another interesting question is "how many errors can be corrected using an efficient list-decoding procedure?"

In this report we will answer the above questions. This report is essentially a survey of two of the main results [GRS00, B02] which look into the above questions.

## 3 Unique Decoding

In this section we look at the algorithmic ideas for unique decoding. We start by looking at the simple algorithm given in table 1. For unique decoding we assume that the number of errors is smaller than $(n-k)/2$.

We now analyse the efficiency and correctness of the algorithm.

INPUT: $\bar{r} =< r_1, \ldots, r_n >$.
OUTPUT: The unique message $m \in \mathbb{Z}_K$ such that $P_{A(\bar{r},CRT(m))} \geqslant N/E$.
PARAMETERS: $E < \sqrt{\frac{N}{K-1}}$

1. Find the $r \in \mathbb{Z}_N$ that uniquely maps to $< r_1, \ldots, r_n >$ by the CRT.
2. Find integers $y, z$ with the following properties:

$$1 \leqslant y \leqslant E, \quad 0 \leqslant z \leqslant N/E \quad \textit{and} \quad y \cdot r \equiv z \ (mod\ N)$$

3. Output $z/y$ if it is an integer.

**Algorithm 1:** Algorithm for unique decoding

**Efficiency** We will show that the algorithm is efficient and then sketch the proof of correctness. The efficiency of step 1 comes from the definition of the CRT. Step 2 is essentially *Integer Linear Programming* in fixed number of variables which has been shown to be efficient (see [L83]).

**Correctness** We will show the correctness of the algorithm under the condition that

$$P_{D(\bar{r},CRT(m))} < E, \tag{1}$$

where the value of $E$ will be decided later. The proof of correctness is broken into two parts. First we show that there exists integers $y, z$, $1 \leqslant y \leqslant E, 0 \leqslant z \leqslant N/E$ such that $y \cdot r \equiv z \ (mod\ N)$. Secondly, given such a pair of integers $y, z$ we will show that $y \cdot m = z$, thus showing that $z/y$ is the correct answer for $m$. The following claims show the two parts.

**Claim 5.** *There exists integers $y, z$, $1 \leqslant y \leqslant E, 0 \leqslant z \leqslant N/E$ such that $y \cdot r \equiv z \ (mod\ N)$.*

*Proof.* We will fix $1 \leqslant y \leqslant E$ and then show the existence of a $z$ satisfying the conditions of the claim. For $D = D(\bar{r}, CRT(m))$, let $y = P_D$, then we have $y \leqslant E$ (from 1). Also, note that $y \cdot r \equiv y \cdot m \ (mod\ p_i)$ for all $i$ (this comes from the simple fact that for any $i \in D, y \cdot r \ (mod\ p_i) = 0 = y \cdot m \ (mod\ p_i)$ and for any $i \in [n] - D, y \cdot r \equiv y \cdot m \ (mod\ p_i)$). So, from CRT we get that $y \cdot m \equiv z \ (mod\ N)$, but since $y \leqslant E, m \leqslant (K-1)$ and $E < \sqrt{N/(K-1)}$, we have that $y \cdot m < N/E$ and hence $z < N/E$. □

**Claim 6.** *Given $1 \leqslant y \leqslant E, 0 \leqslant z \leqslant N/E$ such that $y \cdot r \equiv z \ (mod\ N)$, then $y \cdot m = z$.*

*Proof.* Let $A = A(\bar{r}, CRT(m))$, then for any $i \in A$ we have $z \ (mod\ p_i) = y \cdot r \ (mod\ p_i) = y \cdot m \ (mod\ p_i)$. So, from CRT we get that $y \cdot m \equiv z \ (mod\ P_A)$. From (1) we know that $P_A > N/E$. Also, since $y \leqslant E, m \leqslant (K-1)$ and $E < \sqrt{N/(K-1)}$, $y \cdot m < N/E$. $z$ by definition $\leqslant N/E$ so we get that $y \cdot m = z$. □

The above claims show that the algorithm 1 can uniquely decode a message $m$ given that the amplitude of agreement $P_{A(\bar{r},CRT(m))} \geqslant N/E$. To translate it to the number of errors $e$ that can be corrected in the worst case, we need to consider the case when all the $e$ errors happen in the last $e$ coordinates. Since $E^2 \leqslant N/K$, we get

$$\prod_{i=n-e+1}^{n} p_i^2 \leqslant \prod_{i=n-k+1}^{n} p_i$$

which gives

$$e \leqslant (n-k) \cdot \frac{\log p_1}{\log p_1 + \log p_n}$$

.

# 4 List Decoding

## 4.1 A Simple List-Decoding Algorithm

The first list-decoding algorithm that we will look at in this section is due to [GRS00] and is a simple generalization of the unique decoding algorithm. We define the algorithm and sketch the proof.

INPUT: $\bar{r} =< r_1, \ldots, r_n >$.
OUTPUT: The list of messages $m_1, \ldots, m_l \in \mathbb{Z}_K$ such that $\forall i, P_{A(\bar{r}, CRT(m_i))} \geqslant 2(l+1)F$
PARAMETERS: $F = 2^{\frac{l+1}{2}} \cdot \sqrt{l+1} \cdot N^{\frac{1}{l+1}} \cdot K^{\frac{l+1}{2}}$, $l = \lceil \sqrt{\frac{2n \log p_n}{k \log p_1}} - 1 \rceil$

1. Find the $r \in \mathbb{Z}_N$ that uniquely maps to $< r_1, \ldots, r_n >$ by the CRT.
2. Find integers $c_0, \ldots, c_l$ with the following properties:

$$\forall i, |c_i| \leqslant \frac{F}{K^i}, \quad \sum_i c_i r^i \equiv 0 \ (mod\ N) \quad and \quad < c_0, \ldots, c_l > \neq \bar{0}. \tag{2}$$

3. Output all the roots of the polynomial $\sum_i c_i x^i$.

**Algorithm 2:** Algorithm for list decoding

We now analyse the efficiency and correctness of the algorithm.

**Efficiency** Step 1 is trivial. We work with the assumption that there is an algorithm for step 3 which runs in time $O(l^3(\log F)^3)$ and avoid going into the details for the purpose of this report. For step 2 we construct a lattice and use LLL to get an approximate shortest vector and then use this vector to obtain the $c_i$'s which satisfy (2). Following $(l+2) \times (l+2)$ matrix denotes the basis for the lattice:

$$B = \begin{pmatrix} K^0 & 0 & 0 & \ldots & 0 & 0 \\ 0 & K^1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & K^2 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \ldots & K^l & 0 \\ M \cdot r^0 & M \cdot r^1 & M \cdot r^2 & \ldots & M \cdot r^l & M \cdot N \end{pmatrix}$$ , value of $M$ will be decided later

A general lattice vector in this lattice is given by:

$$\begin{pmatrix} d_0 \cdot K^0 \\ d_1 \cdot K^1 \\ \vdots \\ d_l \cdot K^l \\ M \cdot \left( \sum_{i=0}^{l} d_i r^i + e \cdot N \right) \end{pmatrix} \quad \text{for integer } d_i\text{'s and } e$$

We will show that there is a lattice vector $v$ such that the last coordinate of $v$ is 0 and $||v||$ is small. This is done by the following argument: given $d_i$'s are allowed to take certain large but bounded integers, there exist specific $d_i$'s such such that for these values of $d_i$'s we have $\sum_i d_i r^i \equiv 0 \ (mod\ N)$. We can then use the upper bounds to get a bound on the shortest vector.

More specifically, let $d_i \leqslant N^{\frac{1}{l+1}} \cdot K^{\frac{l+1}{2}-i}$, then consider the function $f(d_0, \ldots, d_n) = \sum_i d_i r^i \ (mod\ N)$. Since the size of the domain of this function ($> N$) is larger than the range ($N$) there are integers $a_0, \ldots, a_n$ such that

$$< a_0, \ldots, a_n > \neq \bar{0} \quad and \quad \forall i, a_i \leqslant N^{\frac{1}{l+1}} \cdot K^{\frac{l+1}{2}-i} \quad and \quad f(a_0, \ldots, a_n) \equiv 0 \ (mod\ N)$$

this implies that $\sum_i a_i r^i = e \cdot N$ for some integer $e$. Let us bound the size of the lattice vector corresponding to these integers. We get that the last coordinate of the lattice vector is 0 and all the other coordinates is $\leqslant N^{\frac{1}{l+1}} \cdot K^{\frac{l+1}{2}}$. This gives us that there is a lattice vector with size at most $\sqrt{l+1} \cdot N^{\frac{1}{l+1}} \cdot K^{\frac{l+1}{2}}$. Since the LLL algorithm always returns a lattice vector of size at most $2^{\frac{l+1}{2}}$ times the size of the shortest vector, the algorithm is guaranteed to return a lattice vector of size at most $F = 2^{\frac{l+1}{2}} \cdot \sqrt{l+1} \cdot N^{\frac{1}{l+1}} \cdot K^{\frac{l+1}{2}}$. So, if we set $M$ to be some integer $> F$ then the LLL algorithm returns a lattice vector with the last coordinate 0 and also gives integer $d_i$'s such that $\sum_i d_i r^i = 0 \ (mod\ N)$ and $|d_i \cdot K^i| \leqslant F$.

**Correctness** Here we need to show that any message $m \in \mathbb{Z}_K$ that has an amplitude of agreement with $\bar{r} \geqslant 2(l+1)F$ will be a root of the polynomial $C(x) = \sum_i d_i x^i$. Let us first get an upper bound on the maximum absolute value that the polynomial $C(x)$ can take. We have $C(x) \leqslant \sum_i |d_i K^i| \leqslant (l+1) \cdot F$. Now, let $A = A(\bar{r}, CRT(m))$ such that $P_A \geqslant 2(l+1)F$. Note that for any $i \in A$ we have $C(r) \ (mod\ p_i) = C(m) \ (mod\ p_i) = 0$. So, from CRT we get that $C(r) \equiv C(m) \equiv 0 \ (mod\ P_A) \equiv 0 \ (mod\ 2(l+1)F)$, but since $C(m) \leqslant (l+1)F$, we get that $C(m) = 0$.

We can optimize the choice of the list size $l$ to minimize the errors. The following theorem from [GRS00] defines these parameters. We omit the proof in this report.

**Theorem 7** ([GRS00])**.** *The Algorithm 2 with parameter* $l = \lceil \sqrt{\frac{2n \log p_n}{k \log p_1}} - 1 \rceil$ *solves the error-correction problem for*

$$e < n - \sqrt{2(k+3)n \frac{\log p_n}{\log p_1}} - \frac{k+6}{2} < n - \sqrt{2kn \frac{\log p_n}{\log p_1}}.$$

### 4.2 A Better List-Decoding Algorithm

Here we look at the list-decoding algorithm due to Boneh [B02] which tolerates errors upto

$$n - \sqrt{kn \frac{\log p_n}{\log p_1}}$$

thus improving upon [GRS00]. The algorithm uses ideas from the Coppersmith's algorithm, though the basic idea remains the same. The essential idea (as also in the previous subsection) is to construct a polynomial $C(x)$ such that all messages $m$ which have high amplitude of agreement $P_A$ with the given corrupted code $\bar{r} =< r_1, \ldots, r_n >$ satisfies $C(m) \equiv 0 \ (mod \ P_A^a)$, for some constant $a$ ($a$ was 1 in the previous subsection). This in conjunction with the fact that for any message $m$, $C(m) < P_A^a$ (since the message space is small), gives that $C(m) = 0$ for those messages that have high amplitude of agreement. Finally, these messages are obtained by finding the roots of the polynomial $C(x)$.

[B02] deviates from [GRS00] in the definition of the amplitude of agreement. We will see later how this helps us in defining the polynomial $C(x)$. Given a corrupted code $\bar{r} =< r_1, \ldots, r_n >$ and a message $m \in \mathbb{Z}_K$, the amplitude of agreement is defined as:

$$M = gcd(N, m - r)$$

where $r \in \mathbb{Z}_N$ such that $\forall i, r \equiv r_i \ (mod \ p_i)$. Note, the definition does not change from [GRS00] if the $p_i$'s are prime. If not, then $M$ might have certain prime factors of $p_i$ even when $m \not\equiv r \ (mod \ p_i)$.

Now, we define the construction of the polynomial. Consider the following sequence of polynomials:

$$\begin{aligned} g_i(x) &= N^{a-i} \cdot (x - r)^i \quad for \ i = 0, \ldots, a - 1 \\ h_i(x) &= (x - r)^a \cdot x^i \quad for \ i = 0, \ldots, a' - 1 \end{aligned}$$

where $a$ and $a'$ will be decided later to optimize the bounds. Note that for any message $m$:

$$\begin{aligned} g_i(m) &\equiv 0 \ (mod \ M^a) \quad for \ i = 0, \ldots, a - 1 \\ h_i(m) &\equiv 0 \ (mod \ M^a) \quad for \ i = 0, \ldots, a' - 1 \end{aligned}$$

So any polynomial $C(x)$ which is a linear combination of the above polynomials satisfies $C(m) \equiv 0 \ (mod \ M^a)$. Any such polynomial has degree at most $d = a + a'$. In addition, if we can somehow ensure that $||C(Kx)|| < M^a/\sqrt{deg(C)}$ [1] , then a simple argument shows that $C(m) = 0$. To obtain a linear combination of polynomials, we can construct a lattice with all these polynomials and then consider a lattice vector. We can then try to find a short lattice vector to satisfy the second requirement. Following is the description of the lattice (it represents the polynomials evaluated at $Kx$, so the lattice vector that we will find will be interpreted as $C(Kx)$).

$$B = \begin{pmatrix} N^a & -N^{a-1} \cdot r & \ldots & N \cdot (-r)^{a-1} & r^a & 0 & \ldots \\ 0 & N^{a-1}K & \ldots & N \cdot \binom{a-1}{a-2}(-r)^{a-2}K & \binom{a}{a-1}(-r)^{a-1}K & r^aK & \ldots \\ 0 & 0 & \ldots & N \cdot \binom{a-1}{a-3}(-r)^{a-3}K^2 & \binom{a}{a-2}(-r)^{a-2}K^2 & \binom{a}{a-1}(-r)^{a-1}K^2 & \ldots \\ \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \ldots \\ 0 & 0 & \ldots & NK^{a-1} & \binom{a}{1}(-r)K^{a-1} & \binom{a}{2}(-r)^2K^{a-1} & \ldots \\ 0 & 0 & \ldots & 0 & K^a & \binom{a}{1}(-r)K^a & \ldots \\ 0 & 0 & \ldots & 0 & 0 & K^{a+1} & \ldots \\ \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \ldots \\ 0 & 0 & \ldots & 0 & 0 & 0 & \ldots \end{pmatrix}$$

[1] for a polynomial $||\sum_{i=0}^{d} c_i x_i||^2 = \sum_{i=0}^{d} c_i^2$

Since the above matrix is an upper triangular matrix, the determinant is simply the product of the diagonal which is

$$det(B) = N^{a(a+1)/2} \cdot K^{d(d-1)/2}$$

We run the LLL algorithm on this lattice to obtain a short lattice vector which we interpret as $C(Kx)$. Now, LLL guarantee us a vector with norm bounded by $2^{d/2}det(B)^{1/d}$. So, we get that

$$||C(Kx)|| < 2^{d/2}det(B)^{1/d} = 2^{d/2}N^{a(a+1)/2d} \cdot K^{(d-1)/2}$$

So, if

$$M^a/\sqrt{d} > 2^{d/2}N^{a(a+1)/2d} \cdot K^{(d-1)/2}$$

then the roots of $C(x)$ give the messages which have amplitude of agreement $> M$. Choosing $a$ appropriately we get that $M > N^{\epsilon}$, where $\epsilon = \sqrt{\frac{\log 4B}{\log P}} + \frac{5}{4d}$. Using the bound on the amplitude of agreement, we can compute the number of errors $e$ that can be corrected in the worst-case which is $e > n - \sqrt{kn\frac{\log p_n}{\log p_1}}$.

## 5 Further Questions

CRT code differs from conventional codes in using alphabet size that is not uniform. We used a uniform notion of *amplitude of agreement* while decoding, as a result of which the amount of error we can decode is conditioned on the locations where these errors happen. Note that we considered the worst case to get a bound on $e$ but this is not really the worst case, as when restricted to places where the message agrees with the word, there is a high amplitude of agreement. If we can implement some kind of *weighted update* method to locate the positions which yield high amplitude of agreement then we can correct more errors. Indeed, [GSM00] use similar ideas to uniquely decode errors upto $e \leqslant (n-k)/2$.

Another way to look at the results is that the techniques shown in this report yields bounds which depend on the ratio $\frac{\log p_n}{\log p_1}$. This means that the higher the discrepancy in the alphabet size, the worst these bounds become. As pointed out earlier, since we are already using a uniform notion for decoding, it should not be too difficult to remove the $\frac{\log p_n}{\log p_1}$ factor. Again, [GSM00] achieves this and give list-decoding algorithm which decodes upto $n - \sqrt{k(n+\epsilon)}$ errors where $\epsilon > 0$ is an arbitrary small constant.

Finally, looking at the recent progress in list-decoding of variants of Reed-Solomon codes, it will be interesting to look into the possibility of extending some of the ideas to CRT decoding.

## References


[B02] D. Boneh.: Finding Smooth Integers in Short Intervals Using CRT Decoding. Journal of Computer and System Sciences (JCSS), Vol. 64, pp. 768–784, 2002.

[GRS00] O. Goldreich, D. Ron, M. Sudan.: Chinese Remaindering with Errors. IEEE Trans. on IT 46(4):1330-1338, 2000.

[GSM00] V. Guruswami, A. Sahai, M. Sudan.: "Soft-decision" Decoding of Chinese Remainder Codes. FOCS, 6: 83 –96, 2000.

[L83] H. W. Lenstra.: Integer Programming with a Fixed Number of Variables. Math. Operations Res., vol. 8, pp 538 – 548, 1983.