

1 List decoding and discrete log problem:

The decoding problem of Reed Solomon codes can be reformulated into the problem of curve fitting or polynomial reconstruction. If we are given n points, $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ in \mathbb{F}_q^2 , we want to find all polynomials $f(x)$ of degree d that pass through at least t points. Recall that Sudan gave an algorithm for $t \geq 2\sqrt{nd}$. Cheng and Wan prove that if we decrease t much further (made precise later), then decoding Reed Solomon codes becomes as hard as solving the discrete log problem.

1.1 Index Calculus versus Sudan's algorithm

We use the index calculus technique for \mathbb{F}_q^* where $\mathbb{F}_q = \mathbb{F}_{p^n} = \mathbb{F}_p[x]/(f(x))$. We consider p midsize, which means that $\log p$ and n^α are of comparable size.

We want to use the factor base

$$B = \{g, x, x + 1, \dots, x + p - 1\}$$

$$\langle g \rangle = \mathbb{F}_p^*$$

The first question is how often $h(x)$ of $\deg h \leq n$ factors as

$$h(x) = g^b \prod_{i=0}^{p-1} (x + i)^{a_i}$$

Total number of h 's is $(p^n - 1)$.

The number of h 's which factor is $(p - 1) \binom{n+p-1}{p}$

If p is big, n is small, this is approximately equal to $\frac{p^n}{n!}$ or c^p when p and n are comparable.

Probability of h factoring = $\frac{1}{n!}$.

But we want the polynomial to factor in the factor base in \mathbb{F}_q^* which is weaker than factoring in $\mathbb{F}_p[x]$.

Cheng's Idea: Lets ask what it means to write h as a product of elements in B in the quotient ring?

$$h(x) = g^b \prod_{i=0}^{p-1} (x + i)^{a_i} + f(x)k(x) \tag{1}$$

But this is not useful to us because we don't know k .

However, if $a_i \neq 0$, for some $i = 0, 1, \dots, p - 1$,

$$h(-i) = f(-i)k(-i)$$

So,

$$k(-i) = \frac{h(-i)}{f(-i)}$$

So, though we don't know k , we know many of the values of k at elements of \mathbb{F}_p . This works only when many a_i 's are non-zero.

Question: Reconstruct $k(x)$ given that we know $k(-i)$ for all $i = 0, \dots, p-1$ with $a_i \neq 0$.

$$d = \deg k(x) = \sum a_i - n$$

$$t = \#\{i = 0 \dots p-1 | a_i \neq 0\}$$

Sudan's algorithm produces a list of solutions k to polynomial reconstruction problem $k(-i) = h(-i)/f(-i)$ for t values of i provided $t \geq 2\sqrt{pd}$ and the list has at most $\sqrt{\frac{p}{d}}$ elements.

If,

$$k(-i) = \frac{h(-i)}{f(-i)}, i \in I \subset \{0, 1, \dots, p-1\}, |I| = t$$

We can deduce $k(x)f(x) - h(x)$ vanishes at $x = -i, i \in I$. This only gives us:

$$k(x)f(x) - h(x) = g^b \prod_{i \in I} (x+i)u(x)$$

This is not quite enough. We want to get rid of $u(x)$.

1.2 How to improve the index calculus algorithm

Fix some d . Given $h(x) \neq 0$ and $\deg h < n$.
Try to find all $k(x)$ of degree $\leq d$, such that

$$k(-i) = \frac{h(-i)}{f(-i)}$$

for at least $\lceil 2\sqrt{pd} \rceil$ values of i and check whether $k(x)f(x) - h(x)$ factors as a product of elements in the factor base.

Note that if d is small, it is better to use Berlekamp's algorithm than Sudan's.

Problem: Count how many more factorizations we are going to achieve by this improvement.

$$2\sqrt{pd} < t < p$$

Thus

$$d < \frac{p}{4}$$

List decoding will not succeed if $d > \frac{p}{4}$.

Theorem 1.1. *If $p > \max(g^2, (n-1)^{2+\epsilon})$ and $g \geq (\frac{4}{\epsilon} + 2)(n+1)$ for some $\epsilon > 0$, then $\forall h, \exists k, a_i, b$, satisfying equation 1 with $\sum a_i \leq g$.*

Question: Can this be improved?

Gain: Count the possible a_i with $\sum a_i \leq (n+d)$ and with $\#\{a_i \neq 0\} \geq 2\sqrt{pd}$.
The good h 's are $(g^b) \prod (x+i)^{a_i} \% f$

1.3 Open questions

Consider: $\mathbb{F}_{p^n}^*, \mathbb{F}_{p^n} = \mathbb{F}_p[x]/f(x)$. Let B be the set of monic irreducible polynomials with degree at most b . $h(x) = g^r \prod_{r(x) \in B} r(x)^{\alpha_r} + f(x)k(x)$ $k(z) = \frac{h(z)}{f(z)}$ if $r(z) = 0$ and $\alpha_r > 0$

Sudan's algorithm gives solution for the field in which we are working. Is there a version of Sudan's algorithm taking into account the field where the values are? Can we use it for the discrete log problem?

1.4 Numerical example of Cheng's idea:

Consider the field $\mathbb{F}_{125} = \mathbb{F}_{5^3} = \mathbb{F}_5[x]/(x^3 - x + 2)$. Instead of working with \mathbb{F}_{125}^* , we will work with $G = \mathbb{F}_{125}^*/\mathbb{F}_5^*$. Thus, $|G| = 31 = 124/4$

$G = \langle x \rangle$. Factor Base $B = x, x + 1, x + 2, x + 3, x + 4$. Also, $\deg h(x) \leq 2$
We ask the following questions: Does $h(x)$ factor over B ? How many $h(x)$ factor over B as polynomials? The number of $h(x)$ that factor over B as polynomials are $1(\text{constant}) + 5(\text{elements of } B, \text{linear}) + 5(\text{squares}) + 10(2 \text{ distinct factors}) = 21$.

So there are 10 irreducible polynomials of degree 2.

How many of these h can be expressed as

$$h(x) = c \prod_{i=0}^4 (x+i)^{a_i} + f(x)k(x)$$

with $c, k \in \mathbb{F}_5^*$? $\deg f = 3, \deg h = 2, \sum a_i = 3$.

Example 1: Consider $h(x) = x^2 + 3$. Remember $f(x) = x^3 - x + 2$. The values of $h(-i)/f(-i)$ for $i = 1, 2, 3, 4, 0$ are $2, 2, 4, 2, 4$
The degree of k is 0, and we try value $k = 2$ which works.

Example 2: $h(x) = x^2 + 2$.

The values of $h(-i)/f(-i)$ for $i = 1, 2, 3, 4, 0$ are $4, 1, 2, 4, 1$
Try $k = 4$. We get $h(x) - 4f(x) = (-4)(x+1)^2(x+4)$.

Example 3: $h(x) = x^2 + 3x + 4$.

The values of $h(-i)/f(-i)$ for $i = 1, 2, 3, 4, 0$ are $1, 2, 3, 4, 2$
Try $k = 2$ $h(x) - 2f(x) = -2(x^2)(x+2)$. Here, another interesting observation was that the values $1, 2, 3, 4$ correspond to the values of the polynomial $-x$. So we also get: $h(x) + xf(x) = (x+1)(x+2)(x+3)(x+4)$.