Discrete logarithms using Reed-Solomon codes

D. Augot, F. Morain

2012/05/09

Contents

Reed-Solomon codes

- Fix $S = \{x_1, x_2, \ldots, x_n\} \in \mathbb{F}_q$.
  Evaluation map:

$$\begin{array}{rccc} \mathrm{ev}_S : & \mathbb{F}_q[X] & \to & \mathbb{F}_q^n \\ & f(X) & \mapsto & (f(x_1), \ldots, f(x_n)) \end{array}$$

- Given k, the k-dimensional Reed-Solomon code $RS(S, k)$ is

$$\{c = \mathrm{ev}_S(f(X)) \mid \ f(X) \in \mathbb{F}_q[X], \ \deg f(X) < k\}.$$

- We say that $f(X)$ is at Hamming distance τ from $y = (y_1, \ldots, y_n)$ if

$$|\{i \in [1 \ldots n]| \ f(x_i) \neq= y_i\}| \leq \tau.$$

  or equivalently: $f(X)$ is μ-close to y, with $\mu + \tau = n$

$$|\{i \in [1 \ldots n]| \ f(x_i) = y_i\}| \geq \mu.$$

  (μ matching positions)

Decoding of Reed-Solomon codes

(List-decoding) problem: of $\mathrm{RS}(S, k)$

Given k and τ, and $\mu = n - \tau$. For $y \in \mathbb{F}_q^n$, find

$$F_\tau(y) = \{f(X) \in \mathbb{F}_q[X]; \quad \deg f(X) < k; \quad d(\mathrm{ev}_S\, f(X), y) \leq \tau\}.$$

When $n - \tau = \mu < k$, there are exponentially many solutions: $n - k$ is the covering radius.

Prop. (Unique decoding) Let $\mu \geq \frac{k+n}{2}$. Then, for any $y \in \mathbb{F}_q^n$,

$$|F_\tau(y)| \leq 1.$$

Prop. (List decoding, "Johnson bound") Let $\mu > \sqrt{n(k-1)}$ be given. Then, for any $y \in \mathbb{F}_q^n$,

$$|F_\tau(y)| \leq \text{is "small"},$$

I.e. *constant* or $O(n^2)$, when k/n is constant and n growing.

Hardness of maximum-likelihood decoding

The maximum-likelihood (ML) decisional problem is:

Given a code $C \subset \mathbb{F}_q^n$, given y, given τ, does there exist $c \in C$ such that $d(c, y) \leq \tau$.

- NP-complete for general linear codes (Berlekamp et al. 1978).
- Also for Reed-Solomon codes (Guruswami-Vardy 2005).
- An amusing consequence is that "deciding deep-holes" is hard.

  (Deep-holes are to Reed-Solomon codes what bent functions are to first order Reed-Muller codes: words as far as possible from the code)

The polynomial reconstruction problem was previously recognized hard (Goldreich-Rubinfeld-Sudan 1995-2000).

Coding theory basic questions

All these hardness results do not concern *real-life codes*:

- What about the size of the field ?

  Guruswami-Vardy: alphabet size exponential in n.

- What about the "support set" S ? We would like $S = \mathbb{F}_q^*$ *(cylic codes)*

  Guruswami-Vardy: only a tiny subset of $\mathbb{F}_q^*$ is used.

- What about the rate k/n ?

  Arbitrary

No proofs for cyclic Reed-Solomon codes.

Unsolved related questions

One would like to know for which radius the problem is hard.

- Provide a radius $\tau = \tau(n, k, q)$ such that decoding RS codes up to radius τ is hard.
- Guruswami-Sudan polynomial-time for $\tau \leq n - \sqrt{(k-1)n}$. No proof that it is the hardness threshold.

  Guruswami-Rudra 2006. Some hints that it is.

- Find τ, and onstruct a word y in $\mathbb{F}_q^n$ τ-far from the RS code such that there is many codewords in the ball of radius τ centered in y.

  Justesen-Høholdt 2001,
  BenSasson-Koparty-Radhakrishnan 2006.

Only partial results, for some classes for codes.

Reed-Solomon codes as crypto-objects ?

"Come on, $\mathbb{F}_{2^8}$ is so small".

- Actually, dealing with $RS_q(n,k)$ may be a big deal.
- q^k can be cryptographically large.

  The standard $RS_{256}(255,k)$ *code has size* 2^{8k}.

- When the realm of computer-algebra style algorithms is left i.e. $\tau > n - \sqrt{(k-1)n}$, no efficient decoding.
- Difficult to trapdoor. Even Generalised Reed-Solomon codes.

  Sidelnikov-Chestakov 1992.

Some efforts : A.-Finiasz 2003, Kiayias-Yung 2000-
Many uses for other primitives: secret-sharing, proof of retrievability, etc.

Cheng-Wan line of work

- Connection between the decoding problem Reed-Solomon codes over $\mathbb{F}_q$, and the DLP in $\mathbb{F}_q^h$.
- More standard codes.
- Weaker hardness result.
  Complexity for discrete logarithm over finite fields, with $x = \mathcal{Q} = \left|\mathbb{F}_{q^h}\right|$

  $$L_x[\alpha, c] = \exp(c(\log x)^\alpha (\log\log x)^{1-\alpha}).$$

  So-called "sub-exponential"

  $$\begin{cases} \text{polynomial} & \alpha = 0 \\ \text{exponential} & \alpha = 1 \end{cases}$$

  Standard $\alpha = 1/2$, better is $\alpha = 1/3$.

Results

- 2004: Reduction in randomized polynomial time (in q) of DLP in $\mathbb{F}_{q^h}$ to the ML-decoding of a standard RS code $[q, k, \_]_q$.

  In particular $k \leq \sqrt{q} - h$. *Vanishing rates.*

- 2010: No algorithm polynomial (in q) for the DLP over $\mathbb{F}_{q^{2h}}$, with $h \leq q^{0.4}$.

  $$\Downarrow$$

  No polynomial time ML-decoding for the standard RS code $[q, k(q)]$, where

  $$\sqrt{q} \leq k(q) \leq q - \sqrt{q}.$$

  Any rate $k/q \in (0, 1)$.

From a “factoring” problem to a decoding problem

- Consider $\mathbb{F}_{q^h}$ a field extension,
- $Q(X) \in \mathbb{F}_q[X]$ is monic irreducible, with $\deg Q(X) = h$,
- $\mathbb{F}_{q^h} = \mathbb{F}_q[X]/Q(X) = \mathbb{F}_q[\overline{X}]$.
- Let $S \subset \mathbb{F}_q$ have size $n \leq q$.

Proposition

There exists $A \subset S$, $|A| = \mu > h$, such that

$$f(X) \equiv \prod_{a \in A} (X - a) \bmod Q(X)$$

if and only if the word

$$y = \mathrm{ev}_S \left(-f(X)/Q(X) - X^k \right)$$

is exactly at distance $\tau = n - \mu$ from the Reed-Solomon code $\mathrm{RS}(S, k)$ of dimension $k = \mu - h$.

Proof

- Suppose that there exists $A \subset S$, $|A| = \mu$, such that
$$\prod_{a \in A} (X - a) \equiv f(X) \bmod Q(X).$$
- There exists $t(X) \in F[X]$, $\deg t(X) = \mu - h = k$, such that
$$\prod_{a \in A} (X - a) = f(X) + t(X)Q(X).$$
- Writing $t(X) = X^k + r(X)$, with $\deg r(X) < k$:
$$\prod_{a \in A} (X - a) = f(X) + (X^k + r(X))Q(X),$$
$$r(X) = -\frac{f(X)}{Q(X)} - X^k + \frac{\prod_{a \in A}(X - a)}{Q(X)},$$
thus $r(a) = -f(a)/Q(a) - a^k$, for $a \in A$.
- Since $|S| = \mu$, $y = \mathrm{ev}_S\left(-f(X)/Q(X) - X^k\right)$ is at distance $n - \mu$ from $\mathrm{ev}_S(r(X)) \in \mathrm{RS}(S, k)$.

Where is the discrete logarithm problem ?

Suppose that $\overline{X}$ is the basis for the logarithms.

- When $f(X) \equiv X^u \bmod Q(X)$, an equation

$$\prod_{a \in A} (X - a) = f(X) \equiv X^u \bmod Q(X) \tag{1}$$

  with $A \subset S$, is called a *relation*.
- Then (1) gives a relation between the logs:

$$\sum_{a \in A} \log(\overline{X} - a) = u \bmod (q^h - 1).$$

- Collecting $n = |S|$ such relations gives a linear system, among whose solutions are the $\log(\overline{X} - a)$.

Still ! Where is the discrete logarithm problem ?

- When all the $\log(\overline{X} - a)$, for $a \in S$ are known, then finding the logarithm of a particular $f(\overline{X})$ can be done by considering

$$\overline{X}^u f(\overline{X})$$

for a random u and trying to find a decomposition

$$\prod_{a \in A} (X - a) \equiv f(X) X^u \bmod Q(X) \tag{2}$$

which gives

$$\log(f(\overline{X})) = \sum_{a \in A} \log(\overline{X} - a) - u$$

- Repeat with random u's until a decomposition (2) if found.

Reed-Solomon based index calculus: First phase

Auxiliary $S \subset \mathbb{F}_q$, $|S| = n$.

1. (Randomize) Compute $f(X) \leftarrow X^u \bmod Q(X)$ for a *random* $u \in \mathbb{Z}/(q^h-1)\mathbb{Z}$.
2. (Decompose-Decode) Find a subset $A \subset S$, $|A| = \mu$, such that
$$f(X) \equiv \prod_{a \in A}(X-a) \bmod Q(X).$$
3. If it exists, add the line
$$u \equiv \sum_{a \in A} \log(\overline{X}-a) \bmod (q^h-1).$$
to a linear system with unknowns the $\log(\overline{X}-a)$.
4. If we have less than n relations, goto 1.
5. (Linear algebra) solve the $n \times n$ linear system over $\mathbb{Z}/(q^h-1)\mathbb{Z}$, which yields the $\log c(\overline{X})$, $c(X) \in S$.

If not full rank, goto to 1 to get new relations.

Reed-Solomon based index calculus: Second phase

Second phase (online): "target" is $\zeta = z(\overline{X})$,

1. (Decompose-decode) find u and $A \subset S$ such that

$$z(X)X^u \equiv \prod_{c \in A} \log(X - a) \bmod Q(X)$$

2. Then $\log z(\overline{X}) \equiv -u + \sum_{c \in A} \log c(\overline{X}) \bmod (q^h - 1)$.

Typical complexity analysis

- 1st phase takes

$$O(n \cdot (1/\pi) \cdot n^{\delta}) + O(n^{\nu}),$$

  $\pi =$ probability of successful decomposition
  $n^{\delta} =$ cost of testing/finding a decomposition
  $n^{\nu} =$ linear algebra

- 2nd phase takes $O((1/\pi) \cdot n^{\delta})$.
- Goal: find parameters to minimize the total time.

Example: Adleman (1/2)

- Consider $S = \{P(X) \in \mathbb{F}_q[X], \text{irreducible of degree} \leq e\}$,

$$n = |S| \approx \frac{q^{e+1}}{e}$$

- We have to consider the probability π that a random polynomial of degree $\leq D$ has all its factors in S:

$$\pi = \frac{N_q(D, e)}{q^D}$$

where $N_q(D, e)$ is

$$|\{P \in \mathbb{F}_q[X], \deg(P) \leq D, \text{all factors of } P \text{ have degree } \leq e\}|.$$

- **Thm.** $\pi \approx (D/e)^{-(1+o(1))D/e}$ (if D and e grow together).

Adleman (2/2)

- Let δ and ν be the exponents for factorization and linear algebra. The cost is:

$$O(n \cdot n^{\delta}/\pi) + O(n^{\nu}).$$

- Balance the costs:

$$(\nu - (\delta + 1)) \log n = -\log \pi.$$

- Using

$$\log B_e \approx e \log q, \quad \log \pi \approx -(D/e) \log(D/e)$$

leads to

$$(\nu - (\delta + 1)) e \log q = \frac{D}{e} \log \frac{D}{e}.$$

- Some workout gives $e = cD^{\alpha}(\log D)^{\beta}$, with $\alpha = \beta = 1/2$.
- Complexity then is

$$\exp(c\sqrt{h \log q \log(h \log q)}) = L_{q^h}[1/2, c]$$

Cheng/Wan in a direct way

1. Use known decoding algorithms of Reed-Solomon codes pour the general framework;
2. We do not pretend at providing a ML-decoding of Reed-Solomon codes;
3. Approaching it for $k/n \to 1$?

   Galand-Fontaine 2009 (for steganography).

Use a device for beaking discrete logarithms over $\mathbb{F}_{2^h}$:

- Xilinx ISE Software. Reed-Solomon Decoder v8.0.

  1 input symbol /clock cycle for $\mathbb{F}_{256}$.

- Aha G709D-40 40 Gbits/sec $[255, 239, \_]$ Reed-Solomon Decoder Core

  $\approx 2 \times 10^7$ *decodings/sec.*

Algorithms for unique decoding

We have a "computer algebra view" of Reed-Solomon codes.

- "Berlekamp-Welch": $O(n^3)$;
- Key equation: $O(n^2)$. Berlekamp-Massey, or EEA (Sugiyama et al.);
- Gao, EEA.

We have chosen Gao's algorithm, which appears to us the easiest to connect to "fast algorithms" from computer algebra.

e.g. von zur Gathen and Gerhard's Computer algebra.

Our aim: find the error-locator polynomial

$$\tau(X) = \prod_{y_i \neq f(a_i)} (X - a_i)$$

or, equivalently,

$$\mu(X) = \prod_{y_i = f(a_i)} (X - a_i).$$

We do not care about the "message polynomial".

Gao1a: basic version

Input : $(x_i) \in \mathbb{F}_q^n$, $(y_i) \in \mathbb{F}_q^n$, k, and thus $d = n - k + 1$.

Precomp. Compute $G(X) = \prod_{i=1}^{n}(X - x_i)$.

Output the error locator polynomial $\tau(x)$ or `failure`.

1. (Interpolation) Compute $I(X)$ such that $I(x_i) = y_i$ for all i.
2. (Partial gcd) Perform PartialEEA with inputs
   $s_0 = G \div X^k$ (of degree $d - 1$),
   $s_1 = I \div X^k$ (of degree $\leq d - 2$)
   Stop when
   $$g(X) = u(X)s_0(X) + v(X)s_1(X)$$
   has $\deg(g) < (d-1)/2$.
3. (Division) Compute $r(X) = G(X) \operatorname{rem} v(X)$
4. If $r(X) = 0$, return $\tau(X) = v(X)$, else return `failure`.

Complexity analysis of Gao1a

Total time is

$$T_G + T_{G \div X^k} + T_{I \div X^k} + T_{PEEA} + T_{v|G?},$$

Rem. Faster version Gao1a useful when $d \ll n$, which is our case; also faster when almost all decoding attempts have to fail!

We need an algorithm which fails fast.

Numerical example I

- Consider $\mathbb{F}_{13^3} = \mathbb{F}_{13}[X]/(X^3 + 2X + 11)$. The support is $S = \{0, 1, \ldots, 12\}$.
- We use $\mathbb{F}_{13}$, and $(n, k, d) = (13, 7, 10)$, which gives $\mu = 7$.
- Consider for instance X^{15}. We have to decode the word:

  $$y = \mathrm{ev}_S(-X^{15}/Q(X) - X^7) = (7, 1, 1, 0, 1, 3, 6, 8, 9, 12, 4, 11, 10).$$

- The PartialEEA procedure yields

  $$u(X) = X^2 + 5X + 3,\ v(X) = 5X^3 + 2X^2 + 3,\ g(X) = 7X + 6,$$

  And the polynomial v factors as $(X-3)(X-8)(X-12)$, so that

  $$X^{15}(X-3)(X-8)(X-12) \equiv G(X) \bmod (Q(X), 13).$$

Numerical example II

Write $13^3 - 1 = 2^2 \cdot 3^2 \cdot 61$ (Pohlig-Hellman).
Logarithms modulo 2^2 and 3^2 are easy to compute.
The matrix M modulo 61 is

$$M = \begin{pmatrix}
15 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\
19 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
33 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
40 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
48 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
51 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\
8 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
15 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
25 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
31 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\
36 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
48 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\
14 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
16 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
17 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
22 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
24 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
27 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}$$

A solution is $V = (1\ 3\ 52\ 24\ 57\ 9\ 41\ 54\ 42\ 27\ 41\ 35\ 5\ 36)^T \bmod 61$.
LDPC codes ? Designs ?

Numerical example III

Computing the logarithm of $X^2 + 1$ is done using the relation

$$(X^2 + 1)X \equiv G(X)/((X(X - 2)(X - 8))) \bmod Q(X).$$

and therefore

$$\log(X^2 + 1) = 417,$$

using the Chinese remaindering theorem.
(Note that this is a toy example, the logarithm of $X^2 + 1$ could have been computed in different ways, factoring it over the factor base directly for instance.)

Almost all decoding attempts have to fail ?

Proposition

There exists $A \subset S$, $|A| = \mu$, such that

$$f(X) \equiv \prod_{a \in A} (X - a) \bmod Q(X)$$

if and only if the word

$$y = \mathrm{ev}_S \left(-f(X)/Q(X) - X^k \right)$$

is exactly at distance $n - \mu$ from the Reed-Solomon code $\mathrm{RS}_S(k)$ of dimension $k = \mu - h$ and support S.

> *We have a $[n, k, n - k + 1]$ Reed-Solomon code, and we want to decode it up to radius $n - k - h$.*

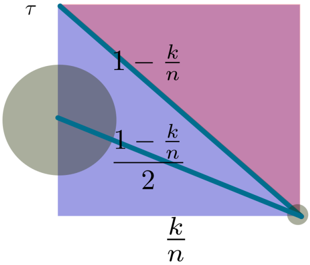
Problem: find S, $n = |S|$, μ.

Density

- Case $n-k$ even. Unique Decoding gives $t=\frac{n-k}{2}$, $\mu=\frac{n+k}{2}$.
- We also have $k=\mu-h$. This gives

$$k=n-2h, \quad \tau=h.$$

- High rate or small rate ?



- Formula for the density

$$\frac{V_q(n,\frac{n-k}{2})\times q^k}{q^n},$$

which is not exponentially small for $k\approx n$.

Oddities

- We look for relations, for $f(X)$ and with $|A| = \mu > h$:

$$f(X) \equiv \prod_{a \in A} (X - a) \bmod Q(X) \tag{3}$$

  It is the RHS $\prod_{a \in A}(X - a)$ which is reduced $\bmod Q(X)$.
- Unique decoding implies no collisions between the

$$\prod_{a \in A} (X - a) \bmod Q(X)$$

- Thus we get a probability of

$$\frac{\binom{n}{\mu}}{q^h} = \frac{\binom{n}{\tau}}{q^h} = \frac{\binom{n}{h}}{q^h}.$$

Analysis

- Recall that the cost is:

$$O\left(n\frac{1}{\varpi}\left(M(n)+M(h)\log h\right)+nhM(h)\log q\right)+O(h\cdot n^2 M(h)).$$

with

$$\pi=\frac{\binom{n}{h}}{\mathcal{Q}}.$$

- For h constant and n going to infinity: $\pi \approx \dfrac{n^{\tau}}{h!\cdot\mathcal{Q}}$.
- If $n > \log q$ and $n > h$, the cost simplifies to

$$O\left(h!(q/n)^h nM(n)\right)+O(h\cdot n^2 M(h)),$$

and the first term always dominates.
- Picking $n=q$, we get

$$O(h!\cdot qM(q)).$$

Sub-exp behaviour ?

- Case $h \ll q$, and growing very slowing, with $\mathcal{Q} = q^h$.
- The cost being $\tilde{O}(h! \cdot q^2)$, we look for $0 \le \alpha < 1$ such that

$$2 \log q + h \log h \simeq c(\log \mathcal{Q})^{\alpha}(\log \log \mathcal{Q})^{1-\alpha},$$

- Making also the hypothesis that $h \ll \log q$ implies

$$\begin{aligned} 2\frac{\log \mathcal{Q}}{h} &\simeq c(\log \mathcal{Q})^{\alpha}(\log \log \mathcal{Q})^{1-\alpha}, \\ h &= \left(\frac{2 \log \mathcal{Q}}{c \log \log \mathcal{Q}}\right)^{1-\alpha} \\ &\simeq \left(\frac{2 \log q}{c \log \log q}\right)^{1/\alpha - 1}. \end{aligned}$$

- To respect the hypothesis $h \ll \log q$, we must have $\alpha \ge 1/2$.

Not my cup of tea...

Conclusion?

Things we do:

- Incremental version: $X^u \to X^{u+1}$ enables to perform incremental decoding, many other tricks.
- Galois actions using extension fields: one can then use $n > q$: this corresponds to codes over extension fields.

Things we may do:

- Use multiplicities to get relations $\prod(X-a)^{e_a}$, with $e_a \in \{1,2\}$.

  Derivative codes (Guruswami-Wang, Beelen), better probabilities, Berlekamp-Welch easy, Gao not so clear (to me).

- Use list decoding at the opposite end of the spectrum $k/n \approx 0$.

Things we dream of:

- Link CRT codes to the case of $\mathbb{F}_p$.
- Elliptic curves. Connection between EC-DLP and decoding of AG codes, for $g = 1$. (Cheng-Wan again).

Theorem–Cheng 2008

For any constant $c > 0$, if there is an algorithm which in expected time $2^{cn}(\log q)^{O(1)}$ computes the minimum distance of any linear $[n, k, \_]_q$ code, then the ECDLP over $\mathbb{F}_q$ can be solved in expected time q^c .

Recall that the generic attack has $c = 1/2$.

Incremental computations

Prop. For u an integer, put
$f(X) = X^u \equiv c_{h-1}X^{h-1} + \cdots + c_0 \bmod Q(X)$ and
$f_1 = X^{u+1} \bmod Q(X)$. Then

$$\frac{f_1(a_i)}{Q(a_i)} = a_i \frac{f(a_i)}{Q(a_i)} - c_{h-1}.$$

Interpolation: $I(a_i) = b_i \to I'(a_i) = b'_i$ with

$$I'(X) = XI(X) + X^{k+1} - X^k + c_{h-1} \bmod G(X).$$

Very easy when $G(X) = X^q - X$.