The Roadmap from Polynomials to Quantum-safe Cryptosystems

A perspective from discrete mathematics (Part 2/4)

Chunlei Li (University of Bergen, Norway)

INCP2-2024/10213 on
Mathematical Theory of Data Transmission and Data Encryption
Oct. 6-10, 2025, Tromsø
Fall school on Geometry in Cryptography and Communication

Outline

1. Representations of Reed-Solomon codes
2. Encoding of RS codes
3. Decoding of RS codes

Representations of Reed-Solomon codes

Background

- introduced by Irving Stoy Reed and Gustave Solomon in 1960
- probably the most widely used error-correcting codes til now
- applications in storage
  - MiniDiscs, CDs, DVDs, Blu-ray discs, QR codes
  - RAID (Redundant Array of Inexpensive Disks) 6
- applications in communications
  - DSL and WiMAX,
  - satellite communications, DVB and ATSC
  - mobile communications
- variants of the generalized RS codes are used in post-quantum cryptography

Representations of RS codes

- represented as BCH codes
- represented by low-degree polynomials

Zeros of Cyclic Codes

- For a cyclic code $\mathcal{C}$ of length n over $\mathbb{F}_q$, its generator polynomials $g(x)|x^n - 1$.
- $g(x)$ can be given by

$$g(x) = m_{i_1}(x) \cdots m_{i_t}(x) = \prod_{j \in C_{i_1}} (x - \xi_n^j) \cdots \prod_{j \in C_{i_t}} (x - \xi_n^j)$$

  where ξ is a primitive n-th root of unity
- Instead of defining a cyclic code $\mathcal{C}$ from by its generator polynomial $g(x)$, one can also define the code $\mathcal{C}$ by the set of all zeros of $g(x)$

Complete Defining Set

- Let $\mathcal{C}$ be a cyclic code of length n over $\mathbb{F}_q$ with a generator polynomial $g(x)$
- Let $\alpha = \xi_n$ be a primitive n-th root of unity in $GF(q^r)$

The **complete defining set** of $\mathcal{C}$ is given by

$$Z(\mathcal{C}) = \{i \in \mathbb{Z}_n : g(\alpha^i) = 0 \text{ for the generator poly. } g(x) \text{ of } \mathcal{C}\}$$

Relation

For a cyclic code $\mathcal{C}$ over $\mathbb{F}_q$ of length n, the relation between its generator polymial and its defining set is given by

$$g(x) = \prod_{i \in Z(\mathcal{C})} (x - \alpha^i).$$

- The complete defining set is a union of disjoint cyclotomic cosets, i.e,, $Z(\mathcal{C}) = C_{i_1} \cup \cdots \cup C_{i_t}$
- **NB**: given a cyclic code $\mathcal{C}$, its (complete) defining set depends on the choice of α
- The dimension of $\mathcal{C}$ is $n - deg(g) = n - |Z(\mathcal{C})|$

Question: what about the minimum distance of $\mathcal{C}$ with prescribed $Z(\mathcal{C})$?

Relation

For a cyclic code $\mathcal{C}$ over $\mathbb{F}_q$ of length n, the relation between its generator polymial and its defining set is given by

$$g(x) = \prod_{i \in Z(\mathcal{C})} (x - \alpha^i).$$

- The complete defining set is a union of disjoint cyclotomic cosets, i.e,, $Z(\mathcal{C}) = C_{i_1} \cup \cdots \cup C_{i_t}$
- **NB**: given a cyclic code $\mathcal{C}$, its (complete) defining set depends on the choice of α
- The dimension of $\mathcal{C}$ is $n - deg(g) = n - |Z(\mathcal{C})|$

Question: what about the minimum distance of $\mathcal{C}$ with prescribed $Z(\mathcal{C})$?

Relation

For a cyclic code $\mathcal{C}$ over $\mathbb{F}_q$ of length n, the relation between its generator polymial and its defining set is given by

$$g(x) = \prod_{i \in Z(\mathcal{C})} (x - \alpha^i).$$

- The complete defining set is a union of disjoint cyclotomic cosets, i.e,, $Z(\mathcal{C}) = C_{i_1} \cup \cdots \cup C_{i_t}$
- **NB**: given a cyclic code $\mathcal{C}$, its (complete) defining set depends on the choice of α
- The dimension of $\mathcal{C}$ is $n - deg(g) = n - |Z(\mathcal{C})|$

Question: what about the minimum distance of $\mathcal{C}$ with prescribed $Z(\mathcal{C})$?

Relation

For a cyclic code $\mathcal{C}$ over $\mathbb{F}_q$ of length n, the relation between its generator polymial and its defining set is given by

$$g(x) = \prod_{i \in Z(\mathcal{C})} (x - \alpha^i).$$

- The complete defining set is a union of disjoint cyclotomic cosets, i.e,, $Z(\mathcal{C}) = C_{i_1} \cup \cdots \cup C_{i_t}$
- **NB**: given a cyclic code $\mathcal{C}$, its (complete) defining set depends on the choice of α
- The dimension of $\mathcal{C}$ is $n - deg(g) = n - |Z(\mathcal{C})|$

Question: what about the minimum distance of $\mathcal{C}$ with prescribed $Z(\mathcal{C})$?

Relation

For a cyclic code $\mathcal{C}$ over $\mathbb{F}_q$ of length n, the relation between its generator polymial and its defining set is given by

$$g(x) = \prod_{i \in Z(\mathcal{C})} (x - \alpha^i).$$

- The complete defining set is a union of disjoint cyclotomic cosets, i.e,, $Z(\mathcal{C}) = C_{i_1} \cup \cdots \cup C_{i_t}$
- **NB**: given a cyclic code $\mathcal{C}$, its (complete) defining set depends on the choice of α
- The dimension of $\mathcal{C}$ is $n - deg(g) = n - |Z(\mathcal{C})|$

Question: what about the minimum distance of $\mathcal{C}$ with prescribed $Z(\mathcal{C})$?

Example

Consider the binary cyclic code $\mathcal{C}$ of length 7 with $Z(\mathcal{C}) = \{1, 2, 4\}$.

Take α as the root of $f(x) = x^3 + x + 1$.

Then $g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^4) = x^3 + x + 1$ is the generator polynomial of $\mathcal{C}$.

$\mathcal{C}$ is thus a binary $[7, 4]$ code (with min. Hamming weight 3)

Minimum Distance of Cyclic Codes

- Computing the true minimum distance of a cyclic code is a hard problem
- The BCH bound is a lower bound for the minimum distance
- Although the BCH bound is tight in many cases, it is not always the true minimum distance

The Bose-Chaudhuri-Hocquenghem (BCH) bound

Let $\mathcal{C}$ be a cyclic code such that its **complete defining set** $Z(\mathcal{C})$ has

$$\delta - 1 \text{ consective elements } b, b+1, \ldots, b+\delta-2,$$

then $\mathcal{C}$ has its minimum distance

$$d(C) \geq \delta.$$

BCH codes

A cyclic code of length n over $\mathbb{F}_q$ with (complete) defining set

$$\{b, b+1, \cdots, b+\delta-2\}$$

is called a BCH code with designed minumum distance δ. Here the generator polynomial is given by

$$g(x) = \text{LeastCommonMultiple}(m_b(x), m_{b+1}(x), \cdots, m_{b+\delta-2}(x)),$$

where $m_i(x)$ is the minimal polynomial of α^i over $\mathbb{F}_q$.

The BCH code

- is called narrow sense if $b = 1$; and
- is called primitive if $n = q^m - 1$ for certain integer m.

BCH view of RS codes

Let α be a primitive element of $\mathbb{F}_q$ and b be an integer. For two positive integers $n \leq q$ and $k < n$, define a polynomial

$$g(x) = (x - \alpha^b)(x - \alpha^{b+1}) \cdots (x - \alpha^{b+n-k-1}).$$

Then the RS code over $\mathbb{F}_q$ of length n is defined as the BCH code with $g(x)$ as its generator polynomial.

For simplicity, we will take $b = 1$.

Parameters of RS codes

- the code length $n \leq q$, and in practice is chosen as $q - 1$
- the minimum distance $d = n - k + 1$
  - the BCH bound gives $d \geq n - k + 1$
  - the Singleton bound gives $d \leq n - k + 1$

Polynomial view of RS codes

Let k be a positive integer. Define a set of polynomials over $\mathbb{F}_q$ as follows

$$\mathcal{P}_k = \{f_0 + f_1 x + \cdots + f_{k-1} x^{k-1} \,:\, f_i \in \mathbb{F}_q\}$$

Let $a_0, \ldots, a_{n-1}$ be n distinct elements in $\mathbb{F}_q$. The (original) RS code is defined by

$$\mathcal{C} = \{(f(a_0), \ldots, f(a_{n-1})) \,:\, f \in \mathcal{P}_k\}\,.$$

Parameters of RS codes

- the code length $n \leq q$, and in practice is chosen as $q - 1$
- the minimum distance $d = n - k + 1$
  - the difference of any two polynomials has degree less than k, so it has at most $k - 1$ zeros, which implies it has at least $n - k + 1$ nonzeros when evaluating at elements $\alpha_1, \ldots, \alpha_n$. This implies $d \geq n - k + 1$
  - the Singleton bound gives $d \leq n - k + 1$

Encoding of RS codes

BCH view

$$m(x) = \sum_{i=0}^{k-1} m_i x^i \longrightarrow c(x) = \sum_{i=0}^{n-1} c_i x^i$$

A codeword is a polynomial of degree $< n$.

With the generator polynomial $g(x) = \prod_{i=1}^{n-k} (x - \alpha^i)$, there are two encoding methods for cyclic RS codes:

1. **normal encoding**: $c(x) = g(x)m(x)$
2. **systematic encoding**: $c(x) = x^{n-k}m(x) - r(x)$ where

$$r(x) \equiv x^{n-k}m(x) \mod g(x)$$

Polynomial/Sequence view

$$(m_0, \ldots, m_{k-1}) \longrightarrow (c_0, \ldots, c_{n-1})$$

A codeword is a sequence of the evaluations of a polynomial at different elements a_j's in $\mathbb{F}_q$.

1. **normal encoding**:
   - $m(x) = m_0 + m_1 x + \cdots + m_{k-1} x^{k-1}$
   - calculate $c_j = m(a_j)$ for $0 \leq j < n$
2. **systematic encoding**
   - use Lagrange interpolation to derive a polynomial $p(x)$ such that $p(a_j) = m_j$ for $j = 0, 1, \ldots, k-1$
   - calculate $c_j = p(a_j)$ for $k \leq j < n$

Decoding of RS codes

Summary of Decoding BCH codes

There are many algorithms for decoding RS codes in the view of BCH codes. The most common ones follow this general outline:

1. Calculate the syndromes s for the received vector
2. Determine the error locator polynomial $\sigma(x)$ from the key equation
3. Determine the error locations $i_1, \cdots, i_t$ from the roots of the error location polynomial
4. Determine the error values $e_{i_1}, \cdots, e_{i_t}$ at those error locations
5. Correct the errors

During some of these steps, the decoding algorithm may determine that the received vector has too many errors, yeilding a decoding failure

Syndrome Calculation

Let $\mathcal{C}$ be a BCH code of length n with defining set

$$I = \{b, b+1, \cdots, b+\delta-2\}.$$

Regard a received word r as $r(x) = \sum_i r_i x^i = c(x) + e(x)$, where $e(x)$ is the error polynomial

Calculate the syndrome $s_j = r(\alpha^j)$ for $j \in I$.

- If $s_j = 0$ for all $j \in I$, then $r(x) = c(x)$ and there's no error;
- IF $s_j \neq 0$ for certain $j \in I$, then $e(x) \neq 0$ and we need to determine the error polynomial $e(x)$

Determine Error-Locator Polynomial

Suppose the error polynomial $e(x)$ has t errors, namely,

$$e(x) = e_{i_1} x^{i_1} + \cdots + e_{i_t} x^{i_t}$$

with $e_{i_r} \neq 0$ for $r = 1, \cdots, t$.

Let $z_r = \alpha^{i_r}$ and define the error-locator polynomial as

$$\sigma(x) = \prod_{r=1}^{t} (1 - z_r x) = 1 + \sigma_1 x + \cdots \sigma_t x^t$$

For $j \in I = \{b, b+1, \cdots, b+\delta-2\}$,

$$s_j = r(\alpha^j) = e(\alpha^j) = \sum_{r=1}^{t} e_{i_r} z_r^j$$

From the defnition of $\sigma(x)$, one has

$$0 = e_{i_r} z_r^{i+t} \sigma(1/z_r) = (e_{i_r} z_r^{i+t} + \sigma_1 e_{i_r} z_r^{i+t-1} + \cdots + \sigma_t e_{i_r} z_r^i)$$

for $r = 1, \cdots, t$.

Therefore, for integers $i \geq 0$,

$$s_{i+t} + \sigma_1 s_{i+t-1} + \cdots + \sigma_t s_i = 0.$$

The key equations have their origin from Newton's identities.

Consider a binary narrow-sense BCH code and verify the key equation, where $s(x) = s_1 + s_2 x + \cdots + s_{2t} x^{2t-1}$,

$$s(x)\sigma(x) \equiv \Omega(x) \mod (x^{2t})$$

where $\Omega(x)$ has degree at most t.

Consider t consective equations

$$\begin{aligned} s_{b+t} &= \sigma_1 s_{b+t-1} + \cdots + \sigma_t s_b \\ s_{b+t+1} &= \sigma_1 s_{b+t} + \cdots + \sigma_t s_{b+1} \\ &\vdots \\ s_{b+2t-1} &= \sigma_1 s_{b+2t-2} + \cdots + \sigma_t s_{b+t-1} \end{aligned}$$

This is the same as

$$\begin{pmatrix} s_{b+t-1} & s_{b+t-2} & \cdots & s_b \\ s_{b+t} & s_{b+t-1} & \cdots & s_{b+1} \\ \vdots & \vdots & \vdots & \vdots \\ s_{b+2t-1} & s_{b+2t-2} & \cdots & s_{b+t-1} \end{pmatrix} \begin{pmatrix} \sigma_1 \\ \sigma_2 \\ \vdots \\ \sigma_t \end{pmatrix} = \begin{pmatrix} s_{b+t} \\ s_{b+t+1} \\ \vdots \\ s_{b+2t-1} \end{pmatrix}$$

Solving the above equation gives the coefficients $\sigma_1, \cdots, \sigma_t$.

The error-locator polynomial $\sigma(x) = 1 + \sum_{i=1}^{t} \sigma_t x^t$ is thus obtained

Determine error-positions

From the definition of $z_r = \alpha^{i_r}$ and

$$\sigma(x) = 1 + \sigma_1 x + \cdots \sigma_t x^t = \prod_{r=1}^{t} (1 - z_r x).$$

In order to determine $\{i_1, \cdots, i_t\}$, we need to use

- brute-force search on the roots $\{z_1, \cdots, z_t\}$ of $\sigma(x)$
- Chien's search on the roots $\{z_1, \cdots, z_t\}$ of $\sigma(x)$

This gives the error positions in the error polynomial $e(x)$

Determine error values

The error values $e_{i_1}, \cdots, e_{i_t}$ can be determined by solving the equations

$$s_j = e(\alpha^j) = e_{i_1} z_1^j + \cdots + e_{i_t} z_t^j$$

for $j \in I = \{b, b+1, \cdots, b+\delta-2\}$.

An more efficient method is to use Forney's algorithm:

- calculate error-evaluation poly. $\Omega(x) = S(x)\sigma(x) \pmod{x^{\delta-1}}$, where $S(x) = s_b + s_{b+1}x + \cdots + s_{b+\delta-2}x^{\delta-2}$
- the error value at position i_r is given by

$$e_{i_r} = \frac{\alpha^{i_k}\Omega(\alpha^{-i_k})}{\alpha^{bi_k}\sigma'(\alpha^{-i_k})}$$

Overview of Decoding RS codes

So far RS codes are the most attractive algebraic codes in applications in the sense that

- there is no restrictions on the parameters n, k, d of RS codes over $\mathbb{F}_q$ except that $n \leq q$
- the nice algebraic structure of RS codes allows for several **polynomial-time** decoding algorithms
  - Peterson-Gorenstein-Zierler alg., Berlekamp-Massey alg., Welch-Berlekamp alg., Sugiyama alg., Gao alg.

Decoding of RS codes in polynomial representation

Given the known parameters of a RS code, including

- the finite field $\mathbb{F}_q$
- parameters $[n, k]$
- the evaluation points $a_0, \ldots, a_{n-1}$

and a received word

$$\mathbf{r} = (r_0, \ldots, r_{n-1}),$$

find the polynomial $f(x)$ of degree $< k$ that gives a codeword

$$\mathbf{c} = (f(a_0), \ldots, f(a_{n-1}))$$

that is closest to $\mathbf{r}$.

Berlekamp-Welch Decoding

- US patent (link) by Berlekamp and Welch in 1983
- created for the original view of RS codes and can be extended for GRS codes

Decoding Strategy

- Suppose the received word **r** contains t errors at positions $i_1, \ldots, i_t$ and set a monic **error polynomial** $E(x)$ with $E(a_{i_r}) = 0$ for $r = 1, \ldots, t$. Then,

$$(f(a_i) - b_i)E(a_i) = 0 \text{ for } b_i = r_i,\ i = 0, 1, \ldots, n-1.$$

- Define $Q(x) = f(x)E(x)$. Then $\deg(Q) \leq k - 1 + t$ and

$$Q(a_i) = b_i E(a_i) \text{ for } 0 \leq i < n.$$

- Take coeff. of $Q(x)$ and $E(x)$ as variables, one has $k + t + t$ variables in n equations. Solve these equation to obtain $E(x)$ and $Q(x)$ and then $f(x) = Q(x)/E(x)$

Decoding Strategy

- Suppose the received word **r** contains t errors at positions $i_1, \ldots, i_t$ and set a monic **error polynomial** $E(x)$ with $E(a_{i_r}) = 0$ for $r = 1, \ldots, t$. Then,

$$(f(a_i) - b_i)E(a_i) = 0 \text{ for } b_i = r_i,\ i = 0, 1, \ldots, n-1.$$

- Define $Q(x) = f(x)E(x)$. Then $\deg(Q) \leq k - 1 + t$ and

$$Q(a_i) = b_i E(a_i) \text{ for } 0 \leq i < n.$$

- Take coeff. of $Q(x)$ and $E(x)$ as variables, one has $k + t + t$ variables in n equations. Solve these equation to obtain $E(x)$ and $Q(x)$ and then $f(x) = Q(x)/E(x)$

Decoding Strategy

- Suppose the received word **r** contains t errors at positions $i_1, \ldots, i_t$ and set a monic **error polynomial** $E(x)$ with $E(a_{i_r}) = 0$ for $r = 1, \ldots, t$. Then,

$$(f(a_i) - b_i)E(a_i) = 0 \text{ for } b_i = r_i,\ i = 0, 1, \ldots, n-1.$$

- Define $Q(x) = f(x)E(x)$. Then $\deg(Q) \leq k - 1 + t$ and

$$Q(a_i) = b_i E(a_i) \text{ for } 0 \leq i < n.$$

- Take coeff. of $Q(x)$ and $E(x)$ as variables, one has $k + t + t$ variables in n equations. Solve these equation to obtain $E(x)$ and $Q(x)$ and then $f(x) = Q(x)/E(x)$