Code based cryptography

Cryptographic Engineering

Clément Pernet

M2 Cyber Security,
UFR-IM^{2}AG, Univ. Grenoble-Alpes
ENSIMAG, Grenoble INP

Outline

Motivation: Post-Quantum Cryptography

Problem (Order finding problem)

Given $a \in \mathbb{Z}_{>0}$ *coprime with* $N \in \mathbb{Z}_{>0}$ *find the smallest* $r \in \mathbb{Z}_{>0}$ *s.t.*

$$a^r = 1 \mod N.$$

Theorem (Shor's algorithm)

The Order finding problem can be solved by a quantum computer in time $O(\log^2 N \log\log N)$.

Factorization with a quantum computer

Corollary

Integer factorization can be solved by a quantum computer in time $O(\log^2 N \log\log N)$.

Factorization with a quantum computer

Corollary

Integer factorization can be solved by a quantum computer in time $O(\log^2 N \log\log N)$.

Sketch of proof.

1. Do
2. Sample a random a
3. $r \leftarrow \text{Order}(a, N)$
4. While $(\text{GCD}(a^{r/2} - 1, N) = 1)$

If r is even then $N | (a^{r/2} - 1)(a^{r/2} + 1)$. But $N \nmid (a^{r/2} - 1)$.

Factorization with a quantum computer

Corollary

Integer factorization can be solved by a quantum computer in time $O(\log^2 N \log\log N)$.

Sketch of proof.

1. Do
2. Sample a random a
3. $r \leftarrow \text{Order}(a, N)$
4. While $(\text{GCD}(a^{r/2} - 1, N) = 1)$

If r is even then $N | (a^{r/2} - 1)(a^{r/2} + 1)$. But $N \nmid (a^{r/2} - 1)$.

- Either $N | a^{r/2} + 1$ (with prob $< 1/2$) $\Rightarrow$restart with another a
- Or the $\text{GCD}(n, a^{r/2} - 1)$ reveals a factor of n.

□

Discrete Logarithm with a quantum computer

Corollary

The Discrete logarithm problem can be solved by a quantum computer in time $O(\log^2 N \log\log N)$.

Discrete Logarithm with a quantum computer

Corollary

The Discrete logarithm problem can be solved by a quantum computer in time $O(\log^2 N \log\log N)$.

Sketch of proof.

Find x such that $g^x = y$ in G of order p. Let

$$f: \begin{array}{ccc} \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z} & \to & G \\ (a, b) & \mapsto & g^a y^{-b} \end{array}$$

, a group isomorphism.

Discrete Logarithm with a quantum computer

Corollary

The Discrete logarithm problem can be solved by a quantum computer in time $O(\log^2 N \log\log N)$.

Sketch of proof.

Find x such that $g^x = y$ in G of order p. Let

$$f : \begin{array}{ccc} \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z} & \to & G \\ (a,b) & \mapsto & g^a y^{-b} \end{array}$$, a group isomorphism.

Note: $f^{-1}(1) = \mathbb{Z}/p\mathbb{Z} \times (x, 1)$.

Find (r_1, r_2) s.t. $f((r_1, r_2) \times (a, b)) = 1$.

Discrete Logarithm with a quantum computer

Corollary

The Discrete logarithm problem can be solved by a quantum computer in time $O(\log^2 N \log\log N)$.

Sketch of proof.

Find x such that $g^x = y$ in G of order p. Let

$$\begin{array}{rccc} f: & \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z} & \to & G \\ & (a,b) & \mapsto & g^a y^{-b} \end{array}$$

, a group isomorphism.

Note: $f^{-1}(1) = \mathbb{Z}/p\mathbb{Z} \times (x,1)$.

Find (r_1, r_2) s.t. $f((r_1, r_2) \times (a,b)) = 1$.

Hence $g^{ar_1} y^{-r_2 b} = g^{ar_1 - xbr_2} = 1$.

Discrete Logarithm with a quantum computer

Corollary

The Discrete logarithm problem can be solved by a quantum computer in time $O(\log^2 N \log\log N)$.

Sketch of proof.

Find x such that $g^x = y$ in G of order p. Let

$$f: \begin{array}{ccc} \mathbb{Z}/p\mathbb{Z} \times \mathbb{Z}/p\mathbb{Z} & \to & G \\ (a,b) & \mapsto & g^a y^{-b} \end{array}$$, a group isomorphism.

Note: $f^{-1}(1) = \mathbb{Z}/p\mathbb{Z} \times (x, 1)$.

Find (r_1, r_2) s.t. $f((r_1, r_2) \times (a, b)) = 1$.

Hence $g^{ar_1} y^{-r_2 b} = g^{ar_1 - xbr_2} = 1$.

$\Rightarrow$ recover x from a, b, r_1, r_2.

□

Post-quantum cryptography

Conclusion:

A quantum computer can break all of classical asymmetric crypto (whenever it is capable of dealing with such instances)

Post-quantum cryptography

Conclusion:

A quantum computer can break all of classical asymmetric crypto (whenever it is capable of dealing with such instances)

Still not quite there yet:

- Number of qu-bits available
- Handling noise

Post-quantum cryptography

Conclusion:

A quantum computer can break all of classical asymmetric crypto (whenever it is capable of dealing with such instances)

Still not quite there yet:

- Number of qu-bits available
- Handling noise

But still a threat:

- Fast progresses, huge efforts
- *Harvest now, decrypt later* already happening
  $\Rightarrow$paradigm of Perfect Forward Secrecy

Post-quantum cryptography

Building new schemes based on other computational hardness assumptions

2016: NIST starts a standardization process calling for proposals for asymetric primitives: signatures and encryption schemes.

2020: 7 finalists of the 1st round + 8 alternative candidates

2024: Expected publication of standard

2030: Expected Q-day

Post-quantum cryptography

Building new schemes based on other computational hardness assumptions

2016: NIST starts a standardization process calling for proposals for asymetric primitives: signatures and encryption schemes.

2020: 7 finalists of the 1st round + 8 alternative candidates

2024: Expected publication of standard

2030: Expected Q-day

Main fields

Lattices: *Kyber* (Module learning-with errors), ...

Coding theory: *McEliece* (Goppa codes)

Multivariate systems: *Oil and Vinegar*

But also

Isogenies: *CSIDH*, but no longer *SIDH*

Post-quantum cryptography

Building new schemes based on other computational hardness assumptions

2016: NIST starts a standardization process calling for proposals for asymetric primitives: signatures and encryption schemes.

2020: 7 finalists of the 1st round + 8 alternative candidates

2024: Expected publication of standard

2030: Expected Q-day

Main fields

Lattices: *Kyber* (Module learning-with errors), ...

Coding theory: *McEliece* (Goppa codes)

Multivariate systems: *Oil and Vinegar*

But also

Isogenies: *CSIDH*, but no longer *SIDH*

Outline

Motivation

Coding Theory

Introduction

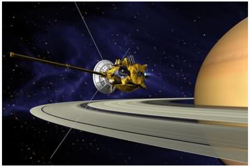
Linear Codes

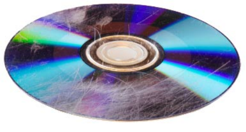
Reed-Solomon codes

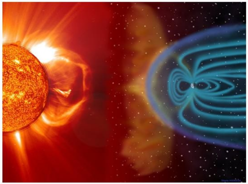
McEliece cryptosystem

Outline

Errors everywhere







Error models

Communication channel

- Radio transmission — electromagnetic interferences
- Ethernet, DSL — electromagnetic interferences
- CD/DVD Audio/Video/ROM — scratches, dust
- RAM — cosmic radiations
- HDD — magnetic field, crash

Error models

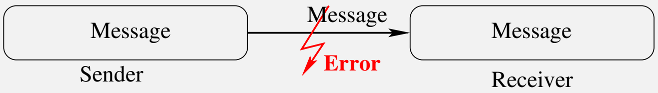
Communication channel

- Radio transmission — electromagnetic interferences
- Ethernet, DSL — electromagnetic interferences
- CD/DVD Audio/Video/ROM — scratches, dust
- RAM — cosmic radiations
- HDD — magnetic field, crash

Error models

Communication channel

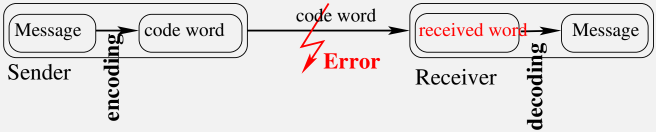
- Radio transmission — electromagnetic interferences
- Ethernet, DSL — electromagnetic interferences
- CD/DVD Audio/Video/ROM — scratches, dust
- RAM — cosmic radiations
- HDD — magnetic field, crash



Error models

Communication channel

- Radio transmission — electromagnetic interferences
- Ethernet, DSL — electromagnetic interferences
- CD/DVD Audio/Video/ROM — scratches, dust
- RAM — cosmic radiations
- HDD — magnetic field, crash



Generalities on coding theory

Goals:

Detect: require retransmission (integrity certificate)

Correct: i.e. when no interraction possible

Tool: Adding redundancy

Generalities on coding theory

Goals:

Detect: require retransmission (integrity certificate)

Correct: i.e. when no interraction possible

Tool: Adding redundancy

Example (NATO phonetic alphabet)

A $\rightarrow$ Alfa, B $\rightarrow$ Bravo, C $\rightarrow$ Charlie, D $\rightarrow$ Delta ...

Alpha Bravo India Tango Tango Echo Delta India Oscar Uniform Sierra !

Generalities on coding theory

Goals:

Detect: require retransmission (integrity certificate)

Correct: i.e. when no interraction possible

Tool: Adding redundancy

Example (NATO phonetic alphabet)

A $\rightarrow$ Alfa, B $\rightarrow$ Bravo, C $\rightarrow$ Charlie, D $\rightarrow$ Delta ...

Alpha Bravo India Tango Tango Echo Delta India Oscar Uniform Sierra !

Two categories of codes:

stream codes: online processing of the stream of information

block codes: cutting information in blocks and applying the same treatment to each block

Generalities on coding theory

Goals:

Detect: require retransmission (integrity certificate)

Correct: i.e. when no interraction possible

Tool: Adding redundancy

Example (NATO phonetic alphabet)

A $\rightarrow$ Alfa, B $\rightarrow$ Bravo, C $\rightarrow$ Charlie, D $\rightarrow$ Delta ...

Alpha Bravo India Tango Tango Echo Delta India Oscar Uniform Sierra !

Two categories of codes:

stream codes: online processing of the stream of information

block codes: cutting information in blocks and applying the same treatment to each block

Generalities and terminology

- A code is a sub-set $\mathcal{C} \subset \mathcal{E}$ of a set of possible words.
- Often, $\mathcal{E}$ is built from an alphabet Σ: $\mathcal{E} = \Sigma^n$.
- Encoding function: $E : \mathcal{S} \rightarrow \mathcal{E}$ such that $E(\mathcal{S}) = \mathcal{C}$.
- A code is
  - t-detector, if any set error on t symbols can be detected
  - t-corrector, if any set error on t symbols can be corrected

Examples

Parity check

$E:(x_1,x_2,x_3)\to(x_1,x_2,x_3,s)$ with $s=\sum_{i=1}^{3}x_i \mod 2 \Rightarrow \sum_{i=1}^{3}x_i+s=0 \mod 2$

Examples

Parity check

$E : (x_1, x_2, x_3) \rightarrow (x_1, x_2, x_3, s) \rightarrow (x_1, x_2, x_3, s) \rightarrow$ Error detected with $s = \sum_{i=1}^{3} x_i \mod 2 \Rightarrow \sum_{i=1}^{3} x_i + s = 0 \mod 2$

Examples

Parity check

$E : (x_1, x_2, x_3) \to (x_1, x_2, x_3, s) \to (x_1, x_2, x_3, s) \to$ Error detected with $s = \sum_{i=1}^{3} x_i \mod 2 \quad \Rightarrow \sum_{i=1}^{3} x_i + s = 0 \mod 2$

Repetition code

- “Say that again?”

Examples

Parity check

$E : (x_1, x_2, x_3) \rightarrow (x_1, x_2, x_3, s) \rightarrow (x_1, x_2, x_3, s) \rightarrow$ Error detected with $s = \sum_{i=1}^{3} x_i \mod 2 \Rightarrow \sum_{i=1}^{3} x_i + s = 0 \mod 2$

Repetition code

- "Say that again?"
- "a" $\rightarrow$ "aaa"$\rightarrow$ "aab"$\rightarrow$ "aaa"$\rightarrow$ "a"

Examples

Parity check

$E : (x_1, x_2, x_3) \to (x_1, x_2, x_3, s) \to (x_1, x_2, x_3, s) \to$ Error detected with $s = \sum_{i=1}^{3} x_i \mod 2 \Rightarrow \sum_{i=1}^{3} x_i + s = 0 \mod 2$

Repetition code

- "Say that again?"
- "a" $\to$ "aaa"$\to$ "aab"$\to$ "aaa"$\to$ "a"

$$E : \begin{array}{ccc} \Sigma & \longrightarrow & \Sigma^r \\ x & \longmapsto & \underbrace{(x, \ldots, x)}_{r \text{ times}} \end{array}$$

, and $\mathcal{C} = \mathrm{Im}(E) \subset \Sigma^r$

Outline

Linear Codes

Linear Codes

Let $\mathcal{E} = V^n$ over a finite field V.
A linear code $\mathcal{C}$ is a subspace of $\mathcal{E}$.

- length: n
- dimension: $k = \dim(\mathcal{C})$
- Rate (of information): k/n

Encoding function: $E : V^k \longrightarrow V^n$ s.t. $\mathcal{C} = \text{Im}(E) \subset \mathcal{V}^n$

Linear Codes

Linear Codes

Let $\mathcal{E} = V^n$ over a finite field V.
A linear code $\mathcal{C}$ is a subspace of $\mathcal{E}$.

- length: n
- dimension: $k = \text{dim}(\mathcal{C})$
- Rate (of information): k/n

Encoding function: $E : V^k \longrightarrow V^n$ s.t. $\mathcal{C} = \text{Im}(E) \subset \mathcal{V}^n$

Example

- Parity code: $k = n - 1$ — 1-detector
- r-repetition code: $k = r/r = 1$ — $r - 1$-detector, $\lfloor \frac{r-1}{2} \rfloor$-corrector

Distance of a code

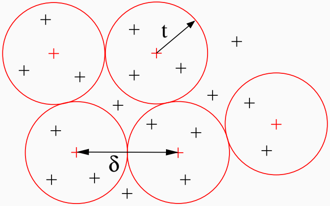
- Hamming weight: $w_H(x) = |\{i, x_i \neq 0\}|$.

Distance of a code

- Hamming weight: $w_H(x) = |\{i, x_i \neq 0\}|$.
- Hamming distance: $d_H(x,y) = w_H(x-y) = |\{i, x_i \neq y_i\}|$

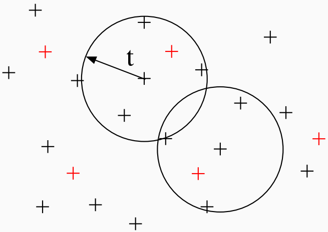
Distance of a code

- Hamming weight: $w_H(x) = |\{i, x_i \neq 0\}|$.
- Hamming distance: $d_H(x, y) = w_H(x - y) = |\{i, x_i \neq y_i\}|$
- Minimum distance of a code $\delta = \min_{x,y \in \mathcal{C}} d_H(x, y)$
  In a linear code: $\delta = \min_{x \in \mathcal{C} \setminus \{0\}} w_H(x))$



Distance of a code

- Hamming weight: $w_H(x) = |\{i, x_i \neq 0\}|$.
- Hamming distance: $d_H(x,y) = w_H(x-y) = |\{i, x_i \neq y_i\}|$
- Minimum distance of a code $\delta = \min_{x,y \in \mathcal{C}} d_H(x,y)$
  In a linear code: $\delta = \min_{x \in \mathcal{C} \setminus \{0\}} w_H(x))$

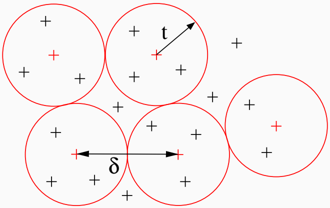


$\mathcal{C}$ is t-corrector if

- $\forall x \in \mathcal{E} \ |\{c \in \mathcal{C}, d_H(x,c) \leq t\}| \leq 1$

Distance of a code

- Hamming weight: $w_H(x) = |\{i, x_i \neq 0\}|$.
- Hamming distance: $d_H(x, y) = w_H(x - y) = |\{i, x_i \neq y_i\}|$
- Minimum distance of a code $\delta = \min_{x,y \in \mathcal{C}} d_H(x, y)$
  In a linear code: $\delta = \min_{x \in \mathcal{C} \setminus \{0\}} w_H(x))$



$\mathcal{C}$ is t-corrector if

- $\forall x \in \mathcal{E}\ |\{c \in \mathcal{C}, d_H(x, c) \leq t\}| \leq 1$
- $\forall c_1, c_2 \in \mathcal{C}\ c_1 \neq c_2 \Rightarrow d_H(c_1, c_2) > 2t$

Perfect codes

Definition

A code is perfect if any detected error can be corrected.

Example

- 4-repetition is not perfect
- 3-repetition is perfect

Perfect codes

Definition

A code is perfect if any detected error can be corrected.

Example

- 4-repetition is not perfect
- 3-repetition is perfect

Property

A code is perfect if the balls of radius t around the codewords form a partition of the ambiant space.

Perfect codes

Definition

A code is perfect if any detected error can be corrected.

Example

- 4-repetition is not perfect
- 3-repetition is perfect

Property

A code is perfect if the balls of radius t around the codewords form a partition of the ambiant space.

Remark

Can be corrected into the wrong code-word. For instance

$(b, a, b) \to (b, b, b)$

Generator matrix and parity check matrix

Generator matrix

- The matrix G of the encoding function (depends on a choice of basis):

$$E : x^T \longrightarrow x^T G$$

- Under systematic form: $G = \left[\begin{array}{ccc|c} 1 & & 0 & \\ & \ddots & & \overline{G} \\ 0 & & 1 & \end{array}\right]$

Generator matrix and parity check matrix

Generator matrix

- The matrix G of the encoding function (depends on a choice of basis):

$$E : x^T \longrightarrow x^T G$$

- Under systematic form: $G = \left[\begin{array}{ccc|c} 1 & & 0 & \\ & \ddots & & \overline{G} \\ 0 & & 1 & \end{array}\right]$

Parity check matrix

1. A matrix $H \in K^{(n-k)\times n}$ such that $\ker(H) = \mathcal{C}$:

$$c \in \mathcal{C} \Leftrightarrow Hc = 0$$

2. A basis of $\ker(G^T)$: $HG^T = 0$

Generator matrix and parity check matrix

Exercise

Find G and H of the parity check and of the k-repetition codes.

Generator matrix and parity check matrix

Exercise

Find G and H of the parity check and of the k-repetition codes.

$$G_{par} = \begin{bmatrix} 1 & & 1 \\ & \ddots & \vdots \\ & 1 & 1 \end{bmatrix}, H_{par} = \begin{bmatrix} 1 & \dots & 1 \end{bmatrix}$$

Generator matrix and parity check matrix

Exercise

Find G and H of the parity check and of the k-repetition codes.

$$G_{par} = \begin{bmatrix} 1 & & 1 \\ & \ddots & \vdots \\ & 1 & 1 \end{bmatrix}, H_{par} = [1 \ \dots \ 1]$$

$$G_{rep} = [1 \ \dots \ 1] = H_{par}, H_{rep} = \begin{bmatrix} 1 & & 1 \\ & \ddots & \vdots \\ & 1 & 1 \end{bmatrix} = G_{par}$$

The parity check code is the ***dual*** *of the repetition code*

Generator matrix and parity check matrix

Exercise

Find G and H of the parity check and of the k-repetition codes.

$$G_{par} = \begin{bmatrix} 1 & & 1 \\ & \ddots & \vdots \\ & 1 & 1 \end{bmatrix}, H_{par} = [1 \; \dots \; 1]$$

$$G_{rep} = [1 \; \dots \; 1] = H_{par}, H_{rep} = \begin{bmatrix} 1 & & 1 \\ & \ddots & \vdots \\ & 1 & 1 \end{bmatrix} = G_{par}$$

The parity check code is the ***dual*** *of the repetition code*

Definition

Let $\mathcal{C}$ be a linear code with generating matrix G and parity check matrix H.

The dual code $\mathcal{D}$ of $\mathcal{C}$ is the linear code with generating matrix H and parity check matrix G.

Role of the parity check matrix

$$c \in \mathcal{C} \Leftrightarrow Hc = 0$$

- Certificate for detecting errors
- Syndrom: $s_x = Hx = H(c+e) = He$

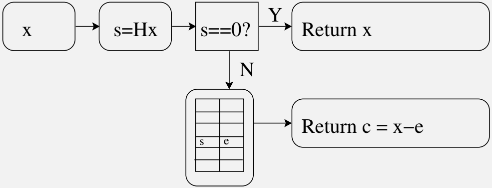
Role of the parity check matrix

$$c \in \mathcal{C} \Leftrightarrow Hc = 0$$

- Certificate for detecting errors
- Syndrom: $s_x = Hx = H(c+e) = He$

A first correction algorithm:

- pre-compute all s_e for $w_H(e) \leq t$ in a table S
- For x received. If $s_x \neq 0$, look for s_x in the table S
- return the corresponding codeword



Decoding problems

Problem (Noisy decoding problem)

Given $G \in K^{k\times n}$ *of full row rank and* $y \in K^n$*, with* $y = c + e$ *where* $c = mG$ *is a code-word and* $w_H(e) = t$*, find* e*.*

Problem (Syndrom decoding problem)

Given $H \in K^{(n-k)\times n}$ *of full row rank and* $s \in K^{n-k}$*, where* $s = He$ *with* $w_H(e) = t$*, find* e*.*

Hamming codes

Let $H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$

- Parameters of the corresponding code?
- Generator matrix?
- Minimal distance?
- Is it a perfect code?

Hamming codes

Let $H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$

- Parameters of the corresponding code? $(n, k) = (7, 4)$

- Generator matrix?

- Minimal distance?

- Is it a perfect code?

Hamming codes

Let $H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$

- Parameters of the corresponding code? $(n, k) = (7, 4)$
- Generator matrix? $G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$
- Minimal distance?
- Is it a perfect code?

Hamming codes

Let $H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$

- Parameters of the corresponding code? $(n, k) = (7, 4)$
- Generator matrix? $G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$
- Minimal distance? $\delta \leq 3$.
  - If $\delta = 1$, $\exists i, H_i = 0$
  - If $\delta = 2$, $\exists i \neq j, H_i = H_j$ $\Rightarrow \delta = 3$
- Is it a perfect code?

Hamming codes

Let $H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$

- Parameters of the corresponding code? $(n, k) = (7, 4)$
- Generator matrix? $G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$
- Minimal distance? $\delta \leq 3$.
  - If $\delta = 1$, $\exists i, H_i = 0$
  - If $\delta = 2$, $\exists i \neq j, H_i = H_j$ $\Rightarrow \delta = 3$
- Is it a perfect code? $\delta = 3$ $\Rightarrow t = 1$ corrector.
  $|\mathcal{C}| = 2^k$ $\Rightarrow \#$ of elements in each ball of radius 1:
  $2^k(1 + 7) = 16 \cdot 8 = 2^7 = |K^n|$ $\Rightarrow$perfect

Hamming codes

Let $H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$

- Parameters of the corresponding code? $(n, k) = (7, 4)$
- Generator matrix? $G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$
- Minimal distance? $\delta \leq 3$.
  - If $\delta = 1$, $\exists i, H_i = 0$
  - If $\delta = 2$, $\exists i \neq j, H_i = H_j \quad \Rightarrow \delta = 3$
- Is it a perfect code? $\delta = 3 \quad \Rightarrow t = 1$ corrector.
  $|\mathcal{C}| = 2^k \quad \Rightarrow \#$ of elements in each ball of radius 1:
  $2^k(1+7) = 16 \cdot 8 = 2^7 = |K^n| \quad \Rightarrow$perfect

Generalization

$\forall \ell$: $H(2^\ell - 1, 2^\ell - \ell)$, is 1-corrector, perfect.
Example: Minitel, ECC memory: $\ell = 7$

Some bounds

Let $\mathcal{C}$ be a code (n, k, δ) over a field $\mathbb{F}_q$ with q elements.

k and δ can not be simulatneously large for a given n.

Sphere packing:

$$q^k \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^n, \text{ with } t = \lfloor \frac{\delta - 1}{2} \rfloor.$$

Some bounds

Let $\mathcal{C}$ be a code (n, k, δ) over a field $\mathbb{F}_q$ with q elements.

k and δ can not be simulatneously large for a given n.

Sphere packing:

$$q^k \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^n, \text{ with } t = \lfloor \frac{\delta - 1}{2} \rfloor.$$

Singleton bound:

$$\delta \leq n - k + 1$$

Some bounds

Let $\mathcal{C}$ be a code (n, k, δ) over a field $\mathbb{F}_q$ with q elements.

k and δ can not be simulatneously large for a given n.

Sphere packing:

$$q^k \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^n, \text{ with } t = \lfloor \frac{\delta - 1}{2} \rfloor.$$

Singleton bound:

$$\delta \leq n - k + 1$$

Sketch of proof:

- Let H be the parity check matrix $(n-k) \times n$.
- δ is the smallest number of linearly dependent cols of H.
- $n - k + 1 = \text{rank}(H) + 1$ cols are always linearly dependent.

Some bounds

Let $\mathcal{C}$ be a code (n, k, δ) over a field $\mathbb{F}_q$ with q elements.

k and δ can not be simulatneously large for a given n.

Sphere packing:

$$q^k \sum_{i=0}^{t} \binom{n}{i} (q-1)^i \leq q^n, \text{ with } t = \lfloor \frac{\delta - 1}{2} \rfloor.$$

Singleton bound:

$$\delta \leq n - k + 1$$

Sketch of proof:

- Let H be the parity check matrix $(n-k) \times n$.
- δ is the smallest number of linearly dependent cols of H.
- $n - k + 1 = \text{rank}(H) + 1$ cols are always linearly dependent.

$\Rightarrow$How to build codes correcting up to $\frac{n-k}{2}$.

Outline

Evaluation-interpolation codes

Theorem (Interpolation)

For all $x_1, \ldots, x_k$*, distincts, and all* $y_1, \ldots, y_k$*, there is a unique polynomial* $f = f_0 + f_1 x + \ldots f_{k-1} x^{k-1}$ *of degree* $< k$ *such that :*

$$f(x_j) = y_j, \quad \textit{for all} \;\; 1 \leq j \leq k.$$

Evaluation-interpolation codes

Theorem (Interpolation)

For all $x_1, \ldots, x_k$*, distincts, and all* $y_1, \ldots, y_k$*, there is a unique polynomial* $f = f_0 + f_1 x + \ldots f_{k-1} x^{k-1}$ *of degree* $< k$ *such that :*

$$f(x_j) = y_j, \quad \text{for all} \quad 1 \leq j \leq k.$$

Corollary

For some fixed x_i*'s*

- *equivalent representation:* $(y_1, \ldots, y_k) \Leftrightarrow (f_0, \ldots, f_{k-1})$.
- *oversampling:* $(y_1, \ldots, y_k, y_{k+1}, \ldots, y_n) \Leftarrow (f_0, \ldots, f_{k-1})$.
  $\Rightarrow$*adding redundancy*

Reed-Solomon codes

Definition (Reed-Solomon codes)

Let K be a finite field, and $x_1, \dots, x_n \in K$ distinct elements. The Reed-Solomon code of length n and dimension k is defined by

$$\mathcal{C}(n,k) = \{(f(x_1), \dots, f(x_n)), f \in K[X]; \deg f < k\}$$

Reed-Solomon codes

Definition (Reed-Solomon codes)

Let K be a finite field, and $x_1, \dots, x_n \in K$ distinct elements. The Reed-Solomon code of length n and dimension k is defined by

$$\mathcal{C}(n,k) = \{(f(x_1), \dots, f(x_n)), f \in K[X]; \deg f < k\}$$

Example

$(n,k) = (5,3), f = x^2 + 2x + 1$ over $\mathbb{Z}/19\mathbb{Z}$.

$(1,2,1,0,0) \xrightarrow{Eval} (f(1), f(5), f(8), f(10), f(12)) = (4,5,17,5,7,17)$

$(4,17,5,7,17) \xrightarrow{Interp.} (1,2,1,0,0)$ $\quad x^2 + 2x + 1$

$(4,17,13,7,17) \xrightarrow{Interp.} (12,8,11,10,1)$ $\quad x^4 + 10x^3 + 11x^2 + 8x + 12$

Minimal distance of Reed-Solomon codes

Property

$\delta = n - k + 1$ *(Maximum Distance Separable codes)*

Minimal distance of Reed-Solomon codes

Property

$\delta = n - k + 1$ *(Maximum Distance Separable codes)*

Proof.

Singeton bound: $\delta \leq n - k + 1$

□

Minimal distance of Reed-Solomon codes

Property

$\delta = n - k + 1$ *(Maximum Distance Separable codes)*

Proof.

Singeton bound: $\delta \leq n - k + 1$

Let $f, g \in \mathcal{C}$: $\deg f, \deg g < k$.

If $f(x_i) \neq g(x_i)$ for $d < n - k + 1$ values x_i,

Then $f(x_j) - g(x_j) = 0$ for at least $n - d > k - 1$ values x_j.

Now $\deg(f - g) < k$, hence $f = g$. $\square$

Minimal distance of Reed-Solomon codes

Property

$\delta = n - k + 1$ *(Maximum Distance Separable codes)*

Proof.

Singeton bound: $\delta \leq n - k + 1$

Let $f, g \in \mathcal{C}$: $\deg f, \deg g < k$.

If $f(x_i) \neq g(x_i)$ for $d < n - k + 1$ values x_i,

Then $f(x_j) - g(x_j) = 0$ for at least $n - d > k - 1$ values x_j.

Now $\deg(f - g) < k$, hence $f = g$. □

⇒correct up to $\frac{n-k}{2}$ errors.

Decoding via the key equation

$$
\begin{aligned}
c &= \mathsf{Eval}(f) && \text{the codeword} \\
y &= c + e && \text{the received word} \\
P &= \mathsf{Interp}((x_i, y_i)) && \text{the interpolant of the received word} \\
\Lambda &= \prod_{i|e_i \neq 0}(x - x_i) && \text{the error locator polynomial}
\end{aligned}
$$

Decoding via the key equation

c	$= \text{Eval}(f)$	the codeword
y	$= c + e$	the received word
P	$= \text{Interp}((x_i, y_i))$	the interpolant of the received word
Λ	$= \prod_{i \mid e_i \neq 0}(x - x_i)$	the error locator polynomial

Property (Key Equation)

$$\Lambda f = \Lambda P \mod \prod_{i=1}^{n}(x - x_i)$$

Decoding via the key equation

c	$= \text{Eval}(f)$	the codeword
y	$= c + e$	the received word
P	$= \text{Interp}((x_i, y_i))$	the interpolant of the received word
Λ	$= \prod_{i \mid e_i \neq 0}(x - x_i)$	the error locator polynomial

Property (Key Equation)

$$\Lambda f = \Lambda P \mod \prod_{i=1}^{n}(x - x_i)$$

Proof.

- correct points: $\forall i$ s.t. $e_i = 0 : f(x_i) = P(x_i)$
- erroneous points: $\forall i$ s.t. $e_i \neq 0 : f(x_i) \neq P(x_i)$ but $\Lambda(x_i) = 0$

Decoding via the key equation

c	$=$ Eval(f)	the codeword
y	$= c + e$	the received word
P	$=$ Interp$((x_i, y_i))$	the interpolant of the received word
Λ	$= \prod_{i \mid e_i \neq 0} (x - x_i)$	the error locator polynomial

Property (Key Equation)

$$\Lambda f = \Lambda P \mod \prod_{i=1}^{n} (x - x_i)$$

Proof.

- correct points: $\forall i$ s.t. $e_i = 0 : f(x_i) = P(x_i)$
- erroneous points: $\forall i$ s.t. $e_i \neq 0 : f(x_i) \neq P(x_i)$ but $\Lambda(x_i) = 0$

$\Rightarrow \forall i \in \{1 \ldots n\} \Lambda(x_i) f(x_i) = \Lambda(x_i) P(x_i)$

$\Rightarrow$ conclusion by C.R.T $\square$

Solving the Key equation

$$\Lambda f = \Lambda P \mod \prod_{i=1}^{n}(x - x_i)$$

- Problem: quadratic equation in the unknwon coeffs of f and Λ

Solving the Key equation

$$N = \Lambda P \mod \prod_{i=1}^{n}(x - x_i)$$

- Problem: quadratic equation in the unknwon coeffs of f and Λ
- Simplifying relaxation: set $N = \Lambda f$ *Linearization*

Berlekamp-Welch decoding

Find N of degree $< k+t$ and Λ monic of degree $\leq t$ s.t.

$$N = \Lambda P \mod \prod_{i=1}^{n}(x - x_i)$$

Linear system solving

$N(X) = n_0 + \ldots n_{k+t-1}X^{k+t-1}$ and $\Lambda(X) = \ell_0 + \cdots + \ell_{t-1}X^{t-1} + X^t$.

Unknonwns: $n_0, \ldots n_{k+t-1}, \ell_0, \ldots, \ell_{t-1}$ ($k+2t$ unknowns)

Equations: each in x_i (n equations)

Berlekamp-Welch decoding

Find N of degree $< k + t$ and Λ monic of degree $\leq t$ s.t.

$$N = \Lambda P \mod \prod_{i=1}^{n}(x - x_i)$$

Linear system solving

$N(X) = n_0 + \ldots n_{k+t-1}X^{k+t-1}$ and $\Lambda(X) = \ell_0 + \cdots + \ell_{t-1}X^{t-1} + X^t$.

Unknonwns: $n_0, \ldots n_{k+t-1}, \ell_0, \ldots, \ell_{t-1}$ ($k + 2t$ unknowns)

Equations: each in x_i (n equations)

$$\left[\begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^{k+t-1} \\ 1 & x_2 & x_1^2 & \ldots & x_1^{k+t-1} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_n^{k+t-1} \end{bmatrix} \middle| \begin{bmatrix} -P(x_1) & & \\ & \ddots & \\ & & -P(x_n) \end{bmatrix} \begin{bmatrix} 1 & x_1 & \ldots & x_1^t \\ \vdots & \vdots & \ldots & \vdots \\ 1 & x_n & \ldots & x_n^t \end{bmatrix}\right] \begin{bmatrix} n_0 \\ \vdots \\ n_{k+t-1} \\ \ell_0 \\ \ldots \\ \ell_{t-1} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

Berlekamp-Welch decoding

Find N of degree $< k + t$ and Λ monic of degree $\leq t$ s.t.

$$N = \Lambda P \mod \prod_{i=1}^{n}(x - x_i)$$

Linear system solving

$N(X) = n_0 + \dots n_{k+t-1}X^{k+t-1}$ and $\Lambda(X) = \ell_0 + \cdots + \ell_{t-1}X^{t-1} + X^t$.

Unknonwns: $n_0, \dots n_{k+t-1}, \ell_0, \dots, \ell_{t-1}$ ($k + 2t$ unknowns)

Equations: each in x_i (n equations)

$$\left[\begin{array}{ccccc|c} 1 & x_1 & x_1^2 & \dots & x_1^{k+t-1} & \\ 1 & x_2 & x_1^2 & \dots & x_1^{k+t-1} & \begin{bmatrix} -P(x_1) & & \\ & \ddots & \\ & & -P(x_n) \end{bmatrix} \begin{bmatrix} 1 & x_1 & \dots & x_1^{t-1} \\ \vdots & \vdots & \dots & \vdots \\ 1 & x_n & \dots & x_n^{t-1} \end{bmatrix} \\ \vdots & \vdots & \vdots & \dots & \vdots & \\ 1 & x_n & x_n^2 & \dots & x_n^{k+t-1} & \end{array}\right] \begin{bmatrix} n_0 \\ \vdots \\ n_{k+t-1} \\ \ell_0 \\ \dots \\ \ell_{t-1} \end{bmatrix} = \begin{bmatrix} P(x_1)x_1^t \\ \vdots \\ P(x_n)x_n^t \end{bmatrix}$$

Rational fraction reconstruction

Problem (RFR: Rational Fraction Reconstruction)

Given $A, B \in K[X]$ *with* $\deg B < \deg A = n$*, find* $f, g \in K[X]$*, such that*

$$\begin{cases} f & = & gB \mod A \\ \deg f & \leq & d_F, \\ \deg g & \leq & n - d_F - 1, \end{cases}.$$

Theorem

Let $(f_0 = A, f_1 = B, \ldots, f_\ell)$ *the sequence of remainders of the extended Euclidean algorithm applied on* (A, B) *and* u_i, v_i *the coefficients s.t.* $f_i = u_i f_0 + v_i f_1$*. Then, at iteration* j *s.t.* $\deg f_j \leq d_F < \deg f_{j-1}$*,*

1. (f_j, v_j) *is a solution of problem RFR.*
2. *it is minimal: any other solution* (f, g) *writes*

$$f = q f_j, \quad g = q v_j \quad \text{for } q \in K[X].$$

Reed-Solomon decoding with Extended Euclidean algorithm

Berlekamp-Welch using extended Euclidean algorithm

- Erroneous interpolant: $P = \text{Interp}((y_i, x_i))$
- Error locator polynomial: $\Lambda = \prod_{i|y_i \text{is erroneous}} (X - x_i)$

Find f with $\deg f \leq d_F$ s.t.. f and P match on $\geq n - t$ evaluations x_i.

$$\underbrace{\Lambda f}_{f_j} = \underbrace{\Lambda}_{g_j} P \mod \prod_{i=1}^{n} (X - x_i)$$

and $(\Lambda f, \Lambda)$ is minimal

$\Rightarrow$computed by extended Euclidean Algorithm

$$f = f_j / g_j.$$

Another decoding algorithm: syndrom based

From now on: $K = \mathbb{F}_q, n = q-1, x_i = \alpha^i$ where α is a primitive n-th root of unity.

$$E(f) = (f(\alpha^0), f(\alpha^1), f(\alpha^2), \ldots, f(\alpha^{n-1})) = DFT_\alpha(f)$$

Another decoding algorithm: syndrom based

From now on: $K = \mathbb{F}_q, n = q-1, x_i = \alpha^i$ where α is a primitive n-th root of unity.

$$E(f) = (f(\alpha^0), f(\alpha^1), f(\alpha^2), \ldots, f(\alpha^{n-1})) = DFT_\alpha(f)$$

Linear recurring sequences

Sequences $(a_0, a_1, \ldots, a_n, \ldots)$ such that

$$\forall j \geq 0 \; a_{j+t} = \sum_{i=0}^{t-1} \lambda_i a_{i+j}$$

generator polynomial: $\Lambda(z) = z^t - \sum_{i=0}^{t-1} \lambda_i z^i$

minimal polynomial: $\Lambda(z)$ of minimal degree

linear complexity of $(a_i)_i$: degree t of the minimal polynomial Λ

Computing Λ_{min}: Berlekamp/Massey algorithm, from $2t$ consecutive

Blahut theorem

Theorem ([Blahut84], [Prony1795])

The DFT$_\alpha$ of a vector of weight t has linear complexity t.

Blahut theorem

Theorem ([Blahut84], [Prony1795])

The DFT$_\alpha$ of a vector of weight t has linear complexity t.

Skecth of proof

- Let $v = e_i$ be a 1-weight vector. Then DFT$_\alpha(v) = \text{Ev}_{(\alpha^0,\alpha^1,\dots,\alpha^n)}(X^i) = ((\alpha^0)^i, (\alpha^1)^i, \dots, (\alpha^{n-1})^i)$ is linearly generated by $\Lambda(z) = z - \alpha^i$.

Blahut theorem

Theorem ([Blahut84], [Prony1795])

The DFT$_\alpha$ of a vector of weight t has linear complexity t.

Skecth of proof

- Let $v = e_i$ be a 1-weight vector. Then $\mathrm{DFT}_\alpha(v) = \mathrm{Ev}_{(\alpha^0,\alpha^1,\ldots,\alpha^n)}(X^i) = ((\alpha^0)^i, (\alpha^1)^i, \ldots, (\alpha^{n-1})^i)$ is linearly generated by $\Lambda(z) = z - \alpha^i$.
- For $v = \sum_{j=1}^t e_{i_j}$, the sequence $\mathrm{DFT}_\alpha(v)$ is generated by $\mathrm{ppcm}_j(z - \alpha^{i_j}) = \prod_{j=1}^t (z - \alpha^{i_j})$

Blahut theorem

Theorem ([Blahut84], [Prony1795])

The DFT$_\alpha$ of a vector of weight t has linear complexity t.

Skecth of proof

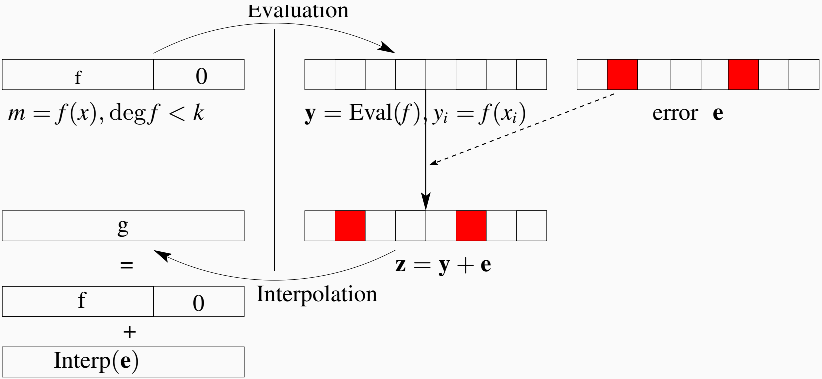
- Let $v = e_i$ be a 1-weight vector. Then $\mathrm{DFT}_\alpha(v) = \mathrm{Ev}_{(\alpha^0,\alpha^1,\ldots,\alpha^n)}(X^i) = ((\alpha^0)^i, (\alpha^1)^i, \ldots, (\alpha^{n-1})^i)$ is linearly generated by $\Lambda(z) = z - \alpha^i$.
- For $v = \sum_{j=1}^{t} e_{i_j}$, the sequence $\mathrm{DFT}_\alpha(v)$ is generated by $\mathrm{ppcm}_j(z - \alpha^{i_j}) = \prod_{j=1}^{t}(z - \alpha^{i_j})$

Corollary

The roots of Λ localize the non-zero elements of v: α^{i_j}.
$\Rightarrow$ *error locator*

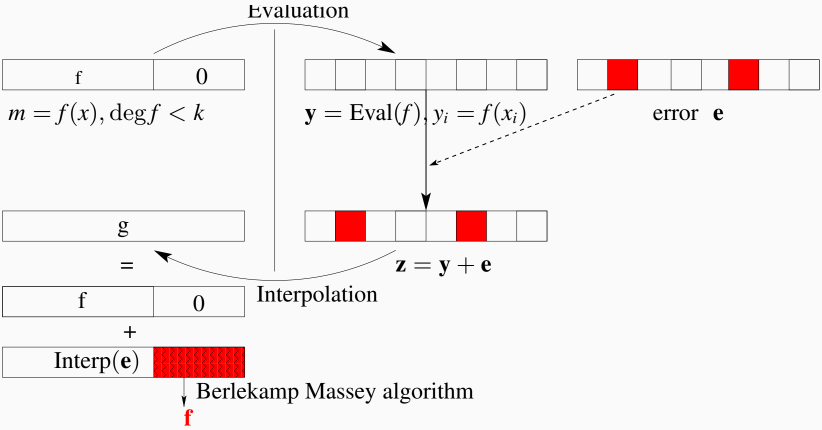
Syndrom Decoding of Reed-Solomon codes

$$\mathcal{C} = \{(f(x_1), \ldots, f(x_n)) \mid \deg f < k\}$$



Syndrom Decoding of Reed-Solomon codes

$$\mathcal{C} = \{(f(x_1), \ldots, f(x_n)) \mid \deg f < k\}$$



Codes derived from Reed Solomon codes

Generalized Reed-Solomon codes

$$\mathcal{C}_{GRS}(n,k,\mathbf{x},\mathbf{v}) = \{(v_1 f(x_1),\ldots,v_n f(x_n)), f \in K_{<k}[X]\}$$

- Same dimension and minimal distance $\Rightarrow$MDS
- Existence of a dual GRS code in the same evaluation points: There is a vector $\mathbf{w}$ such that

$$\mathcal{C}_{GRS}(n,k,\mathbf{x},\mathbf{v})^{\perp} = \mathcal{C}_{GRS}(n,n-k,\mathbf{x},\mathbf{w})$$

  i.e.

$$G_{\text{GRS}}(\mathbf{x},\mathbf{w})G_{\text{GRS}}(\mathbf{x},\mathbf{v})^T = 0$$

(Proof in exercise)

Codes derived from Reed-Solomon

Alternant codes

Motivation: workaround the limitatoin of GRS codes: $n \leq q$

$\Rightarrow$allow for arbitrary length n given a fixed field $\mathbb{F}_q$.

Idea: use a GRS over an extension $\mathbb{F}_{q^m}$, and restrict to $\mathbb{F}_q$.

Let

- $K = \mathbb{F}_q, \bar{K} = \mathbb{F}_{q^m}$ and $\mathbf{x} \in \bar{K}^n$,$\mathbf{v} \in (\bar{K}^*)^n$
- $\mathcal{C}_{\bar{K}} = \mathcal{C}_{GRS}(n, k, \mathbf{x}, \mathbf{v})$ over $\bar{K}$ with minimum distance $D = n - k + 1$

Then

$$\mathcal{C}_{Alt} = \mathcal{C}_{\bar{K}} \cap \mathbb{F}_q^n$$

- Dimension: $\geq n - (D-1)m = n - (n-k)m$
- Minimum distance: $\geq D$ by design

(Proof in exercise)

Codes derived from Reed Solomon codes

Goppa codes

- An instance of a broad class of Algebraic Geometric Codes (AG-codes).
- Can be viewed as an alternant code for some special multiplier vector $\mathbf{v}$.

Let

- $K = \mathbb{F}_q, \bar{K} = \mathbb{F}_{q^m}$ and $\mathbf{x} \in \bar{K}^n$
- $f \in \mathbb{F}_{q^m}[X]$, $\deg f = r$ and $mr < n$
- $\mathbf{v} = (\frac{f(x_i)}{\prod_{j \neq i}(x_j - x_i)})$
- $\mathcal{C}_{\bar{K}} = \mathcal{C}_{GRS}(n, n-r, \mathbf{x}, \mathbf{v})$ over $\bar{K}$ with parameters $(n, n-r, r+1)$

Then

$$\mathcal{C}_{Goppa} = \mathcal{C}_{\bar{K}} \cap \mathbb{F}_q^n$$

Codes derived from Reed Solomon codes

Goppa codes

- An instance of a broad class of Algebraic Geometric Codes (AG-codes).
- Can be viewed as an alternant code for some special multiplier vector $\mathbf{v}$.

Let

- $K = \mathbb{F}_q, \bar{K} = \mathbb{F}_{q^m}$ and $\mathbf{x} \in \bar{K}^n$
- $f \in \mathbb{F}_{q^m}[X]$, $\deg f = r$ and $mr < n$
- $\mathbf{v} = (\frac{f(x_i)}{\prod_{j \neq i}(x_j - x_i)})$
- $\mathcal{C}_{\bar{K}} = \mathcal{C}_{GRS}(n, n-r, \mathbf{x}, \mathbf{v})$ over $\bar{K}$ with parameters $(n, n-r, r+1)$

Then

$$\mathcal{C}_{Goppa} = \mathcal{C}_{\bar{K}} \cap \mathbb{F}_q^n$$

- Dimension: $\geq n - rm$
- Minimum distance: $\geq r + 1$
- Case $q = 2^e$, with f square free
  $\Rightarrow$ Minimum distance: $= 2r + 1$

Outline

A code based cryptosystem [Mc Eliece 78]

Designing a one way function with trapdoor

Use the encoder of a linear code:

$$\text{message} \times \left[G \right] + \text{rand. error} = \text{codeword}$$

Encryption: is easy (matrix-vector product)

Decryption: decoding a received word

- easy for known codes
- NP-complete for random linear codes

Trapdoor: efficient decoding when the code familiy is known

A code based cryptosystem [Mc Eliece 78]

Designing a one way function with trapdoor

Use the encoder of a linear code:

$$\text{message} \times \left[G \right] + \text{rand. error} = \text{codeword}$$

Encryption: is easy (matrix-vector product)

Decryption: decoding a received word

- easy for known codes
- NP-complete for random linear codes

Trapdoor: efficient decoding when the code familiy is known

⇒requires a family $\mathcal{F}$ of codes

- indistinguishable from random linear codes
- with fast decoding algorithm

Mc Eliece Cryptosystem

KeyGen

- Select an (n, k) binary linear code $\mathcal{C} \in \mathcal{F}$ correcting t errors, having an efficient decoding algorithm $\mathcal{A}_{\mathcal{C}}$,
- Form $G \in \mathbb{F}_q^{k \times n}$, a generator matrix for $\mathcal{C}$
- Sample uniformly a $k \times k$ non-singular matrix S
- Select uniformly an n-dimensional permutation P.
- $\hat{G} = SGP$

Public key: $(\hat{G}, t)$

Private key: (S, G, P)

Mc Eliece Cryptosystem

Encrypt

$$E(\mathbf{m}) = \mathbf{m}\hat{G} + \mathbf{e} = \mathbf{m}SGP + \mathbf{e} = \mathbf{y}$$

where $\mathbf{e}$ is an error vector of Hamming weight at most t.

Decrypt

1. $\mathbf{y}' = \mathbf{y}P^{-1} \quad = \mathbf{m}SG + \mathbf{e}P^{-1}$
2. $\mathbf{m}' = \mathcal{A}_{\mathcal{C}}(\mathbf{y}') \quad = \mathbf{m}S$
3. $\mathbf{m} = \mathbf{m}'S^{-1}$

Parameters for Mc Eliece in practice

(n,k,d)	Code family	key size	Security	Attack
$(256,128,129)$	Gen. Reed-Solomon	67ko	2^{95}	[SS92]

Parameters for Mc Eliece in practice

(n, k, d)	Code family	key size	Security	Attack
$(256, 128, 129)$	Gen. Reed-Solomon subcodes of GRS	67ko	2^{95}	[SS92] [Wie10]

Parameters for Mc Eliece in practice

(n, k, d)	Code family	key size	Security	Attack
$(256, 128, 129)$	Gen. Reed-Solomon	67ko	2^{95}	[SS92]
	subcodes of GRS			[Wie10]
$(1024, 176, 128)$	Reed-Muller codes	22.5ko	2^{72}	[MS07, CB13]
$(2048, 232, 256)$	Reed-Muller codes	59.4ko	2^{93}	[MS07, CB13]

Parameters for Mc Eliece in practice

(n, k, d)	Code family	key size	Security	Attack
$(256, 128, 129)$	Gen. Reed-Solomon	67ko	2^{95}	[SS92]
	subcodes of GRS			[Wie10]
$(1024, 176, 128)$	Reed-Muller codes	22.5ko	2^{72}	[MS07, CB13]
$(2048, 232, 256)$	Reed-Muller codes	59.4ko	2^{93}	[MS07, CB13]
$(171, 109, 61)_{128}$	Alg.-Geom. codes	16ko	2^{66}	[FM08, CMP14]

Parameters for Mc Eliece in practice

(n, k, d)	Code family	key size	Security	Attack
$(256, 128, 129)$	Gen. Reed-Solomon	67ko	2^{95}	[SS92]
	subcodes of GRS			[Wie10]
$(1024, 176, 128)$	Reed-Muller codes	22.5ko	2^{72}	[MS07, CB13]
$(2048, 232, 256)$	Reed-Muller codes	59.4ko	2^{93}	[MS07, CB13]
$(171, 109, 61)_{128}$	Alg.-Geom. codes	16ko	2^{66}	[FM08, CMP14]
$(1024, 524, 101)_2$	Goppa codes	67kB	2^{62}	
$(2048, 1608, 48)_2$	Goppa codes	412kB	2^{96}	
$(6960, 5413, 239)_2$	Goppa codes	8MB	2^{128}	

Advantages of McEliece cryptosystem

Security

Based on two assumptions:

- decoding a random linear code is hard (NP complete reduction)
- the generator matrix of a Goppa code, perturbed by S and P looks random (indistinguishability)

Pros:

- faster encoding/decoding algorithms than RSA, ECC (for a given security parameter)
- Post quantum security: still robust against quantum computer attacks

Cons:

- harder to use for signature (non determinstic encoding)
- large key size