Linear Codes and Applications in Cryptography

Matthias Minihold
matthias.minihold@gmx.at

Vortrag an der Ruhr-Universität Bochum

am 17. September 2015

Overview

Chapters

1. Linear Codes

Overview

Chapters

1. Linear Codes
2. Cryptography

RUHR UNIVERSITÄT BOCHUM RUB

Overview

Chapters

1. Linear Codes
2. Cryptography
3. Example of PKS based on Goppa Codes using Sage

Overview

Chapters

1. Linear Codes
2. Cryptography
3. Example of PKS based on Goppa Codes using Sage
4. Quantum Computing

RUHR UNIVERSITÄT BOCHUM RUB

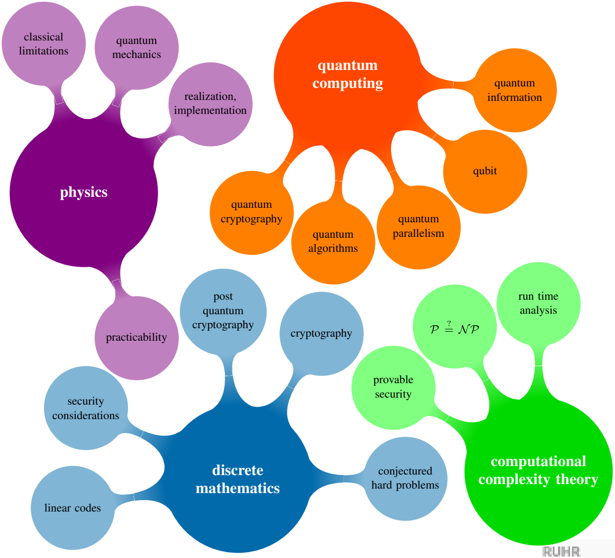

Figure : A mind map visualizing the topics in this thesis.

Linear Codes: Goppa Codes

We define Goppa codes over a general alphabet $\mathbb{F}_q$ and present decoding advantages in the binary case, because of Patterson's algorithm and the larger minimum distance between codewords.

Linear Codes: Goppa Codes

We define Goppa codes over a general alphabet $\mathbb{F}_q$ and present decoding advantages in the binary case, because of Patterson's algorithm and the larger minimum distance between codewords.

Definition

Let $G(z) \in \mathbb{F}_{q^m}[z]$ be a Goppa polynomial of degree $t := \deg G(z)$ and the support $L = \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \subseteq \mathbb{F}_{q^m}$, such that $G(\alpha) \neq 0$, for all $\alpha \in L$. The Goppa code $\Gamma(L, G)$ is defined by:

$$\Gamma(L, G) := \left\{ c \in \mathbb{F}_q^n \mid \sum_{i=1}^{n} \frac{c_i}{z - \alpha_i} \equiv 0 \mod G(z) \right\}.$$

Linear Codes: Goppa Codes

Theorem

Let $G(z) = \sum_{i=0}^{t} g_i z^i$ *with* $g_i \in \mathbb{F}_{q^m}, g_t \neq 0$ *be the Goppa polynomial and let the support be* $L = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$.

RUHR UNIVERSITÄT BOCHUM RUB

Linear Codes: Goppa Codes

Theorem

Let $G(z) = \sum_{i=0}^{t} g_i z^i$ with $g_i \in \mathbb{F}_{q^m}, g_t \neq 0$ be the Goppa polynomial and let the support be $L = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. Then the resulting Goppa code $\Gamma(L, G)$ is a linear code with parameters $[n, k \geq n - mt, d \geq t + 1]$ over $\mathbb{F}_q$.

Linear Codes: Goppa Codes

Theorem

Let $G(z) = \sum_{i=0}^{t} g_i z^i$ *with* $g_i \in \mathbb{F}_{q^m}, g_t \neq 0$ *be the Goppa polynomial and let the support be* $L = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$*.*
Then the resulting Goppa code $\Gamma(L, G)$ *is a linear code with parameters* $[n, k \geq n - mt, d \geq t + 1]$ *over* $\mathbb{F}_q$*.*

Theorem

Given a Goppa polynomial $G(z)$ *over* $\mathbb{F}_2$ *of degree* $t := \deg G(z)$*.*

Linear Codes: Goppa Codes

Theorem

Let $G(z) = \sum_{i=0}^{t} g_i z^i$ with $g_i \in \mathbb{F}_{q^m}, g_t \neq 0$ be the Goppa polynomial and let the support be $L = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. Then the resulting Goppa code $\Gamma(L, G)$ is a linear code with parameters $[n, k \geq n - mt, d \geq t + 1]$ over $\mathbb{F}_q$.

Theorem

Given a Goppa polynomial $G(z)$ over $\mathbb{F}_2$ of degree $t := \deg G(z)$.

- *If G has no multiple zeros and*
- *the lowest degree perfect square $\overline{G}(z)$ that is divisible by $G(z)$ is $\overline{G}(z) = G(z)^2$,*

Linear Codes: Goppa Codes

Theorem

Let $G(z) = \sum_{i=0}^{t} g_i z^i$ with $g_i \in \mathbb{F}_{q^m}, g_t \neq 0$ be the Goppa polynomial and let the support be $L = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$. Then the resulting Goppa code $\Gamma(L, G)$ is a linear code with parameters $[n, k \geq n - mt, d \geq t + 1]$ over $\mathbb{F}_q$.

Theorem

Given a Goppa polynomial $G(z)$ over $\mathbb{F}_2$ of degree $t := \deg G(z)$.

- *If G has no multiple zeros and*
- *the lowest degree perfect square $\overline{G}(z)$ that is divisible by $G(z)$ is $\overline{G}(z) = G(z)^2$,*

then the Goppa code $\Gamma(L, G)$ has minimum distance $d \geq 2t + 1$.

JB

Cryptography: McEliece PKS

Cryptography: McEliece PKS

Algorithm 1: McEliece key generation

Input : $(k \times n)$ generator matrix G, error correcting capability t
Output: public key (G', t), private key (S, G, P)

Choose a $(n \times n)$ permutation matrix P
Choose a regular binary $(k \times k)-$matrix S
Compute $(k \times n)$ matrix $G' = SGP$

Algorithm 2: McEliece encryption

Input : message m, public key (G', t) and thus implicitly n, k
Output: encrypted message c

Compute $c' = mG'$
Randomly generate a vector $z \in \mathbb{F}_q^n$,
with non-zero entries at $\leq t$ positions
Compute $c = c' + z$, the cipher text block

Cryptography: McEliece PKS

Algorithm 3: McEliece decryption

Input : encrypted message block c, private key (S, G, P)
Output: message m

Compute $\overline{c} = cP^{-1}$
The decoding algorithm of the code C corrects t errors. $\overline{c} \to \overline{m}$.
Compute $m = \overline{m}S^{-1}$, the clear text message block.
`// Precompute the matrices` P^{-1} `and` S^{-1} `once.`

RUHR UNIVERSITÄT BOCHUM RUB

Example of code-based PKS: $n = 8, m = 3, q = 2$.

Example (Binary Goppa code $\Gamma(L, G)$)

The degree $t = 2$ Goppa polynomial $G(z) = z^2 + z + 1$ and the support $L = \{0, 1, \beta, \beta^2, \beta + 1, \beta^2 + \beta, \beta^2 + \beta + 1, \beta^2 + 1\}$ yield an $[n = 8, k = 8 - 2 \cdot 3, d = 2 \cdot 2 + 1]$−code $\Gamma(L, G) \leq \mathbb{F}_2^8$.

RUHR UNIVERSITÄT BOCHUM RUB

Example of code-based PKS: $n = 8, m = 3, q = 2$.

Example (Binary Goppa code $\Gamma(L, G)$)

The degree $t = 2$ Goppa polynomial $G(z) = z^2 + z + 1$ and the support $L = \{0, 1, \beta, \beta^2, \beta + 1, \beta^2 + \beta, \beta^2 + \beta + 1, \beta^2 + 1\}$ yield an $[n = 8, k = 8 - 2 \cdot 3, d = 2 \cdot 2 + 1]$−code $\Gamma(L, G) \leq \mathbb{F}_2^8$.

$$G_{Goppa} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$H_{Goppa} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Example of code-based PKS: $n = 8, m = 3, q = 2$.

Example (Binary Goppa code $\Gamma(L, G)$)

The degree $t = 2$ Goppa polynomial $G(z) = z^2 + z + 1$ and the support $L = \{0, 1, \beta, \beta^2, \beta + 1, \beta^2 + \beta, \beta^2 + \beta + 1, \beta^2 + 1\}$ yield an $[n = 8, k = 8 - 2 \cdot 3, d = 2 \cdot 2 + 1]$–code $\Gamma(L, G) \leq \mathbb{F}_2^8$.

$$G_{Goppa} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

$$H_{Goppa} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

RUHR UNIVERSITÄT BOCHUM RUB

McEliece PKS

$$S \cdot G_{Goppa} \cdot P = G_{pub} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

RUHR UNIVERSITÄT BOCHUM RUB

McEliece PKS

$$S \cdot G_{Goppa} \cdot P = G_{pub} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Example (McEliece)

`Alice generates:` $u = (0, 1)$

`Alice sends:` $y = (1, 0, 0, 0, 1, 1, 1, 0)$

McEliece PKS

$$S \cdot G_{Goppa} \cdot P = G_{pub} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}.$$

Example (McEliece)

`Alice generates:` $u = (0, 1)$
`Alice sends:` $y = (1, 0, 0, 0, 1, 1, 1, 0)$
`Bob receives:` $y = (1, 0, 0, 0, 1, 1, 1, 0)$
`y*P^{-1}:` $yP = (0, 0, 1, 0, 0, 1, 1, 1)$
`Bob decodes yD:` $yD = (0, 0, 1, 1, 1, 1, 1, 1)$
`scrambled information bits mm:` $(0, 1)$
`mm*S^{-1}:` $yS = (0, 1)$ The decryption was successful!

Niederreiter PKS

$$M \cdot H_{Goppa} \cdot P = H_{pub} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

RUHR UNIVERSITÄT BOCHUM RUB

Niederreiter PKS

$$M \cdot H_{Goppa} \cdot P = H_{pub} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Example (Niederreiter)

`Alice generates:` $u = (0,0,1,0,0,0,0,1)$

`Alice sends:` $y = (0,1,1,1,1,1)$

Niederreiter PKS

$$M \cdot H_{Goppa} \cdot P = H_{pub} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Example (Niederreiter)

`Alice generates:` $u = (0, 0, 1, 0, 0, 0, 0, 1)$
`Alice sends:` $y = (0, 1, 1, 1, 1, 1)$
`Bob receives:` $y = (0, 1, 1, 1, 1, 1)$
`M^{-1}*y:` $yM = (0, 1, 1, 1, 0, 0)$
`Bob decodes xD:` $xD = (1, 0, 0, 0, 0, 1, 0, 0)$
`P^{-1}*xD:` $xS = (0, 0, 1, 0, 0, 0, 0, 1)$
The decryption was successful

Quantum computing

Definition

In general, a qubit is in the state:
$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle, \quad |a_0|^2 + |a_1|^2 = 1.$

Quantum computing

Definition

In general, a qubit is in the state:
$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle, \quad |a_0|^2 + |a_1|^2 = 1.$

- Classical computer: 1 processor can be used repeating some calculation $\mathcal{O}(2^n)$ times to perform one gate operation on each of the 2^n values representable by n bits.

Quantum computing

Definition

In general, a qubit is in the state:
$|\psi\rangle = a_0 |0\rangle + a_1 |1\rangle, \quad |a_0|^2 + |a_1|^2 = 1.$

- Classical computer: 1 processor can be used repeating some calculation $\mathcal{O}(2^n)$ times to perform one gate operation on each of the 2^n values representable by n bits.
- Quantum computer: 2^n values are representable using n qubits. A quantum gate applied to these n qubit takes $\mathcal{O}(n)$ time.

RUHR UNIVERSITÄT BOCHUM RUB

Quantum computing: Assumptions

Assumption classically	False, quantum mechanically

Quantum computing: Assumptions

Assumption classically	False, quantum mechanically
A bit has a definite value.	A qubit after it is read.

Quantum computing: Assumptions

Assumption classically	False, quantum mechanically
A bit has a definite value.	A qubit after it is read.
A bit can only be 0 or 1.	Superposition of 0 and 1.

Quantum computing: Assumptions

Assumption classically	False, quantum mechanically
A bit has a definite value.	A qubit after it is read.
A bit can only be 0 or 1.	Superposition of 0 and 1.
A bit can be copied without affecting its value.	Copying necessarily changes a qubit's quantum state.

RUHR UNIVERSITÄT BOCHUM RUB

Quantum computing: Assumptions

Assumption classically	False, quantum mechanically
A bit has a definite value.	A qubit after it is read.
A bit can only be 0 or 1.	Superposition of 0 and 1.
A bit can be copied without affecting its value.	Copying necessarily changes a qubit's quantum state.
A bit can be read without affecting its value.	Reading a qubit in a superposition will change it.

RUHR UNIVERSITÄT BOCHUM RUB

Quantum computing: Assumptions

Assumption classically	False, quantum mechanically
A bit has a definite value.	A qubit after it is read.
A bit can only be 0 or 1.	Superposition of 0 and 1.
A bit can be copied without affecting its value.	Copying necessarily changes a qubit's quantum state.
A bit can be read without affecting its value.	Reading a qubit in a superposition will change it.
Reading one bit has no affect on any other (unread) bit.	Entangled qubits: reading one qubit will affect the other.

Table : Assumptions about bits that are not true at the quantum scale.

Post-quantum cryptography: PKS

Post-quantum cryptography: PKS

- The speedup thanks to Shor's quantum algorithm over the best known classical algorithm for Factorization problem is:

$$\mathcal{O}\left(e^{(C+o(1))n^{\frac{1}{3}}(\log n)^{\frac{2}{3}}}\right) \xrightarrow{\text{Shor}} \mathcal{O}(n^3).$$

RUHR UNIVERSITÄT BOCHUM RUB

Post-quantum cryptography: PKS

- The speedup thanks to Shor's quantum algorithm over the best known classical algorithm for Factorization problem is:

$$\mathcal{O}\left(e^{(C+o(1))n^{\frac{1}{3}}(\log n)^{\frac{2}{3}}}\right) \xrightarrow{\text{Shor}} \mathcal{O}(n^3).$$

- The discrete logarithm problem on elliptic curves (ECDLP) is affected, too — with an exponential speedup, where N denotes the number of points on the elliptic curve:

$$\mathcal{O}(\sqrt{N}) = \mathcal{O}\left(e^{\frac{\log N}{2}}\right) \xrightarrow{\text{Shor}} \mathcal{O}((\log\ N)^3).$$

RUHR UNIVERSITÄT BOCHUM RUB

Post-quantum cryptography: PKS

- The speedup thanks to Shor's quantum algorithm over the best known classical algorithm for Factorization problem is:

$$\mathcal{O}\left(e^{(C+o(1))n^{\frac{1}{3}}(\log n)^{\frac{2}{3}}}\right) \xrightarrow{\text{Shor}} \mathcal{O}(n^3).$$

- The discrete logarithm problem on elliptic curves (ECDLP) is affected, too — with an exponential speedup, where N denotes the number of points on the elliptic curve:

$$\mathcal{O}(\sqrt{N}) = \mathcal{O}\left(e^{\frac{\log N}{2}}\right) \xrightarrow{\text{Shor}} \mathcal{O}((\log\ N)^3).$$

- McEliece and Niederreiter PKS are still unbroken, if based on binary Goppa codes.

RUHR UNIVERSITÄT BOCHUM RUB

Post-quantum cryptography: PKS

- The speedup thanks to Shor's quantum algorithm over the best known classical algorithm for Factorization problem is:

$$\mathcal{O}\left(e^{(C+o(1))n^{\frac{1}{3}}(\log n)^{\frac{2}{3}}}\right) \xrightarrow{\text{Shor}} \mathcal{O}(n^3).$$

- The discrete logarithm problem on elliptic curves (ECDLP) is affected, too — with an exponential speedup, where N denotes the number of points on the elliptic curve:

$$\mathcal{O}(\sqrt{N}) = \mathcal{O}\left(e^{\frac{\log N}{2}}\right) \xrightarrow{\text{Shor}} \mathcal{O}((\log\ N)^3).$$

- McEliece and Niederreiter PKS are still unbroken, if based on binary Goppa codes. Generalizations of all known attacks seem unfeasible.

RUHR UNIVERSITÄT BOCHUM RUB

Review - 2 years later

Noteworthy remarks on the thesis after review

Review - 2 years later

Noteworthy remarks on the thesis after review

- Objective of this thesis was the combination of multiple scientific fields to a consistent text about uses of linear codes in cryptography.

Review - 2 years later

Noteworthy remarks on the thesis after review

- Objective of this thesis was the combination of multiple scientific fields to a consistent text about uses of linear codes in cryptography. I chose and suggested the topic.

Review - 2 years later

Noteworthy remarks on the thesis after review

- Objective of this thesis was the combination of multiple scientific fields to a consistent text about uses of linear codes in cryptography. I chose and suggested the topic.
  During implementation one "bleak spot" in the literature appeared — all 7 sources didn't make the maths behind one step in Patterson's Decoding Algorithm explicit.

RUHR UNIVERSITÄT BOCHUM RUB

Review - 2 years later

Noteworthy remarks on the thesis after review

- Objective of this thesis was the combination of multiple scientific fields to a consistent text about uses of linear codes in cryptography. I chose and suggested the topic.
  During implementation one "bleak spot" in the literature appeared — all 7 sources didn't make the maths behind one step in Patterson's Decoding Algorithm explicit.
  I filled this "hole" for decoding Binary Goppa Codes and proved, implemented and demonstrated the functionality.

RUHR UNIVERSITÄT BOCHUM RUB

Review - 2 years later

Noteworthy remarks on the thesis after review

- Objective of this thesis was the combination of multiple scientific fields to a consistent text about uses of linear codes in cryptography. I chose and suggested the topic.
  During implementation one "bleak spot" in the literature appeared — all 7 sources didn't make the maths behind one step in Patterson's Decoding Algorithm explicit.
  I filled this "hole" for decoding Binary Goppa Codes and proved, implemented and demonstrated the functionality.
- Generating Chapter 3 from .tex source computes the examples on the fly with random input (by calling Sage) and checks validity displaying "True" (or "False") within the text!

RUHR UNIVERSITÄT BOCHUM RUB

Review - 2 years later

Noteworthy remarks on the thesis after review

- Objective of this thesis was the combination of multiple scientific fields to a consistent text about uses of linear codes in cryptography. I chose and suggested the topic.
  During implementation one "bleak spot" in the literature appeared — all 7 sources didn't make the maths behind one step in Patterson's Decoding Algorithm explicit.
  I filled this "hole" for decoding Binary Goppa Codes and proved, implemented and demonstrated the functionality.
- Generating Chapter 3 from .tex source computes the examples on the fly with random input (by calling Sage) and checks validity displaying "True" (or "False") within the text!
  Thus I had trust in my implementation.

RUHR UNIVERSITÄT BOCHUM RUB

Thank you for your attention!

References

Thank you for your attention!

Matthias Minihold, Master's Thesis (2013)
Linear Codes and Applications in Cryptography.
Vienna University of Technology.