Attacks in code based cryptography: a survey, new results and open problems

J.-P. Tillich
Inria, team-project SECRET

April 9, 2018

1. Code based cryptography

Difficult problem in coding theory

Problem 1. [Decoding]
Input: n, r, t *with* $r < n$, *parity-check matrix* $\boldsymbol{H} \in \mathbb{F}_q^{r \times n}$, $\boldsymbol{s} \in \mathbb{F}_q^r$
Question: $\exists$? $\boldsymbol{e}$ *such that*

$$\begin{cases} \boldsymbol{H}\boldsymbol{e}^\intercal & = & \boldsymbol{s}^\intercal \\ |\boldsymbol{e}| & \leqslant & t \end{cases}$$

where $|\boldsymbol{e}| =$ *hamming weight of* $\boldsymbol{e} = \#\{i \in [\![1, n]\!], e_i \neq 0\}$.

Problem NP-complete

The dual problem

$$\text{Code } \mathcal{C} \overset{\text{def}}{=} \left\{\boldsymbol{c} \in \mathbb{F}_q^n : \boldsymbol{H}\boldsymbol{c}^\intercal = 0\right\}$$

$$\dim \mathcal{C} = n - r = k$$

Input: t , $\mathcal{C}$ subspace of dim k of $\mathbb{F}_q^n$, $\boldsymbol{y} \in \mathbb{F}_q^n$

Question: $\exists$? $\boldsymbol{c} \in \mathcal{C}$ such that $|\boldsymbol{y} - \boldsymbol{c}| \leqslant t$.

$$\boldsymbol{H}\underbrace{(\boldsymbol{y} - \boldsymbol{c})}_{\boldsymbol{e}}{}^\intercal = \boldsymbol{H}\boldsymbol{y}^\intercal = \boldsymbol{s}^\intercal$$

$\boldsymbol{y}$ = the word that we want to decode

$\boldsymbol{e}$ = $\boldsymbol{y} - \boldsymbol{c} =$ the error we want to find

A long-studied problem

Correct. t errors in a code of length n and dim. k has cost $\tilde{O}(2^{\alpha(\frac{k}{n},\frac{t}{n})n})$

Author(s)	Year	$\max_{R,\tau} \alpha(R,\tau)$
Prange	1962	0.1207
Stern	1988	0.1164
Dumer Bernstein, Lange, Peters	1991 2011	0.1162
May, Meurer and Thomae	2011	0.1114
Becker, Joux, May, Meurer	2012	0.1019
May, Ozerov	2015	0.0966
Both, May	2017	0.0953
Both, May	2018	0.0885

Complexities collapse when $t = o(n)$

- [CantoTorres, Sendrier, 2016] complexity $2^{-\log(1-R)t(1+o(1))}$ when $t = o(n)$ and where $R = k/n$

Code-based cryptography

Code $\mathcal{C} \stackrel{\text{def}}{=} \{\boldsymbol{c} \in \mathbb{F}_q^n : \boldsymbol{H}\boldsymbol{c}^\intercal = 0\}$

- Take a code that has an efficient decoding algorithm
- Public key: random parity-check matrix of the code $\boldsymbol{H}_{\text{rand}} = \boldsymbol{Q}\boldsymbol{H}$ where $\boldsymbol{Q}$ is a random invertible matrix in $\mathbb{F}_q^{r \times r}$
- Private key: trapdoor to the efficient decoding algorithm

Two approaches

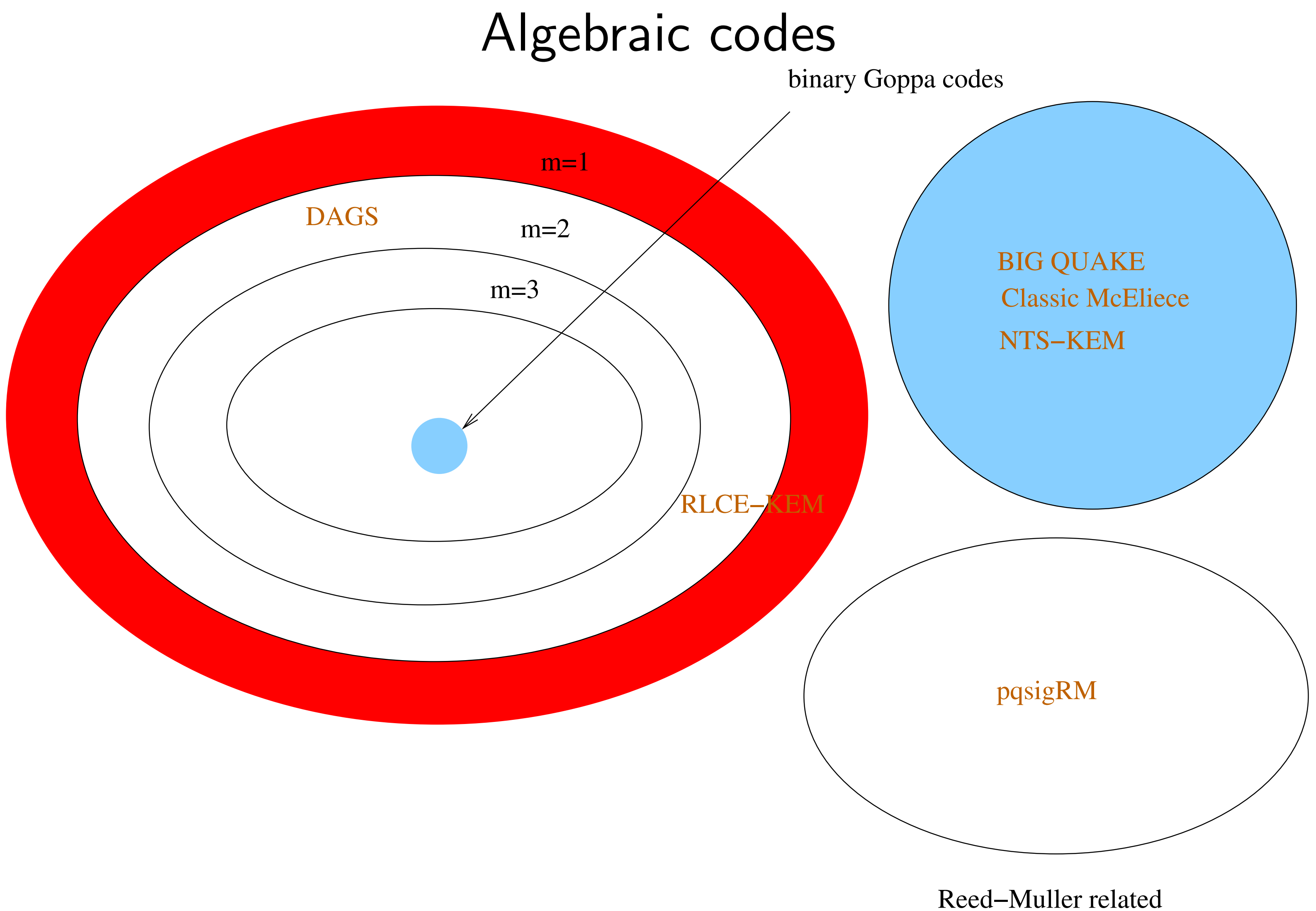
- Pick up your favorite code (that has an efficient decoder)
- Choose a code/scheme with a reduction to decoding a generic linear code

History

- 1978 McEliece: binary Goppa codes
- 1986 Niederreiter variant based on GRS codes
- 1991 Gabidulin, Paramonov, Tretjakov: Gabidulin codes
- 1994 Sidelnikov: Reed-Muller codes
- 1996 Janwa-Moreno: algebraic geometric codes
- 199* a zillion propositions with LDPC codes
- 2003 Alekhnovich: Alekhnovich system
- 2005 Berger-Loidreau: subcodes of GRS codes
- 2006 Wieschebrink, GRS codes + random columns in the generator matrix

- 2008 Baldi-Bodrato-Chiaraluce: LDPC based MDPC codes
- 2010 Bernstein, Lange, Peters: non-binary wild Goppa codes
- 2012 Misoczki-Tillich-Barreto-Sendrier: MDPC codes
- 2012 Löndahl-Johansson: convolutional codes
- 2013 Gaborit, Murat, Ruatta, Zémor: LRPC codes
- 2014 Shrestha, Kim: polar codes
- 2014 Hooshmand, Shooshtari, Eghlidos, Aref: subcodes of polar codes

Code based NIST submissions in Hamming metric



Code based NIST submissions in Hamming metric

Non-algebraic codes

- BIKE
- HQC
- LEDAkem
- LEDApkc
- Lepton
- QC-MDPC
- RaCoSS

Code based NIST submissions in the rank metric

- Edon-K
- LAKE
- LOCKER
- McNie
- Ouroubouros-R
- RankSign
- RQC

2. The main cryptanalytic techniques for attacking the key

- Finding small weight codewords in $\mathcal{C}$ or in $\mathcal{C}^{\perp}$ that reveal the underlying structure
- Algebraic attacks
- Product considerations
- Folding techniques
- Computing the hull $\mathcal{C} \cap \mathcal{C}^{\perp}$

3. Product considerations

Square code attacks

Definition 1. [Componentwise product] *Given two vectors* $\boldsymbol{a} = (a_1, \ldots, a_n)$ *and* $\boldsymbol{b} = (b_1, \ldots, b_n) \in \mathbb{F}_q^n$, *we denote by* $\boldsymbol{a} \star \boldsymbol{b}$ *the componentwise product*

$$\boldsymbol{a} \star \boldsymbol{b} \stackrel{\text{def}}{=} (a_1 b_1, \ldots, a_n b_n)$$

Definition 2. [Product of codes & square code] *The* star product code *denoted by* $\mathcal{A} \star \mathcal{B}$ *of* $\mathcal{A}$ *and* $\mathcal{B}$ *is the vector space spanned by all products* $\boldsymbol{a} \star \boldsymbol{b}$ *where* $\boldsymbol{a}$ *and* $\boldsymbol{b}$ *range over* $\mathcal{A}$ *and* $\mathcal{B}$ *respectively. When* $\mathcal{B} = \mathcal{A}$, $\mathcal{A} \star \mathcal{A}$ *is called the* square code *of* $\mathcal{A}$ *and is rather denoted by* $\mathcal{A}^2$.

Dimension of the square code

$\mathcal{A}$ and $\mathcal{B}$ codes with respective bases $(\boldsymbol{a}_i)$ and $(\boldsymbol{b}_j)$.

1. $\dim(\mathcal{A} \star \mathcal{B}) \leqslant \dim(\mathcal{A}) \dim(\mathcal{B})$ (generated by the $\boldsymbol{a}_i \star \boldsymbol{b}_j$'s)
2. $\dim(\mathcal{A}^2) \leqslant \binom{\dim(\mathcal{A})+1}{2}$ (generated by the $\boldsymbol{a}_i \star \boldsymbol{a}_j$'s with $i \leqslant j$)

Generalized Reed-Solomon (GRS) codes

Definition 3. [Generalized Reed-Solomon code] *Let k and n be integers such that $1 \leqslant k < n \leqslant q$ where q is a power of a prime number. The generalized Reed-Solomon code* $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ *of dimension k is associated to a pair $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{F}_q^n \times \mathbb{F}_q^n$ where $\boldsymbol{x}$ is an n-tuple of distinct elements of $\mathbb{F}_q$ and the entries y_i are arbitrary nonzero elements in $\mathbb{F}_q$.* $\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$ *is defined as:*

$$\mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \stackrel{def}{=} \Big\{(y_1 p(x_1), \dots, y_n p(x_n)) : p \in \mathbb{F}_q[X], \deg p < k\Big\}.$$

$\boldsymbol{x}$ is the support and $\boldsymbol{y}$ the multiplier.

GRS codes, alternant codes

- A GRS code corrects $\frac{n-k}{2}$ errors.

Definition 1. *Let $\boldsymbol{x} \in (\mathbb{F}_{q^m})^n, \boldsymbol{y} \in (\mathbb{F}_{q^m})^n$ be as in the definition of GRS codes. The alternant code $\textbf{Alt}_r(\boldsymbol{x}, \boldsymbol{y})$ is defined by*

$$\textbf{Alt}_r(\boldsymbol{x}, \boldsymbol{y}) \stackrel{def}{=} \underbrace{\textbf{GRS}_r(\boldsymbol{x}, \boldsymbol{y})^{\perp}}_{\textbf{GRS}_{n-r}(\boldsymbol{x}, \boldsymbol{y}')} \cap (\mathbb{F}_q)^n$$

Proposition 1.

$$\begin{aligned} \dim \textbf{Alt}_r(\boldsymbol{x}, \boldsymbol{y}) &\geqslant n - mr \\ d_{min} \textbf{Alt}_r(\boldsymbol{x}, \boldsymbol{y}) &\geqslant r + 1 \end{aligned}$$

What is wrong with generalized Reed-Solomon codes ?

When $\mathcal{C}$ is a random code of length n, with high probability [Cascudo, Cramer, Mirandola, Zémor]

$$\dim(\mathcal{C}^2) = \min \left\{ \binom{\dim(\mathcal{C}) + 1}{2}, n \right\}$$

When $\mathcal{C}$ is a generalized Reed-Solomon code

$$\dim(\mathcal{C}^2) = \min \{ 2 \dim(\mathcal{C}) - 1, n \}$$

The explanation

$$\boldsymbol{c} = (y_1 p(x_1), \ldots, y_n p(x_n)),\ \boldsymbol{c}' = (y_1 q(x_1), \ldots, y_n q(x_n)) \in \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$$

where p and q are two polynomials of degree at most $k-1$.

$$\boldsymbol{c} \star \boldsymbol{c}' = \left(y_1^2 p(x_1) q(x_1), \ldots, y_n^2 p(x_n) q(x_n)\right) = \left(y_1^2 r(x_1), \ldots, y_n^2 r(x_n)\right)$$

where r is a polynomial of degree $\leqslant 2k-2$.

$$\Longrightarrow \boldsymbol{c} \star \boldsymbol{c}' \in \mathbf{GRS}_{2k-1}(\boldsymbol{x}, \boldsymbol{y}^2)$$

The Wieschebrink attack on the Berger-Loidreau cryptosystem

- known: a subcode $C \subset \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$
- unknown: $\boldsymbol{x}$ and $\boldsymbol{y}$.

If the codimension of C is small enough

$$C \star C = \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \star \mathbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) = \mathbf{GRS}_{2k-1}(\boldsymbol{x}, \boldsymbol{y}')$$

The Wieschebrink attack

1. Compute $C \star C = \mathbf{GRS}_{2k-1}(\boldsymbol{x}, \boldsymbol{y}')$
2. Recover $\boldsymbol{x}$ and $\boldsymbol{y}'$ by using the Sidelnikov-Shestakov algorithm.

Filtration attack

[Couvreur, Otmani, T 2014]: Attack on wild Goppa codes when $m = 2$.

A filtration for GRS codes

A new attack on McEliece based on GRS codes.

known : $C_0 = \textbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y})$

unknown : $\boldsymbol{x}$,$\boldsymbol{y}$.

$$C_0 = \textbf{GRS}_k(\boldsymbol{x}, \boldsymbol{y}) \supseteq C_1 = \textbf{GRS}_{k-1}(\boldsymbol{x}, \boldsymbol{y}) \supseteq \cdots \supseteq C_{k-1} = \textbf{GRS}_1(\boldsymbol{x}, \boldsymbol{y})$$

The point:

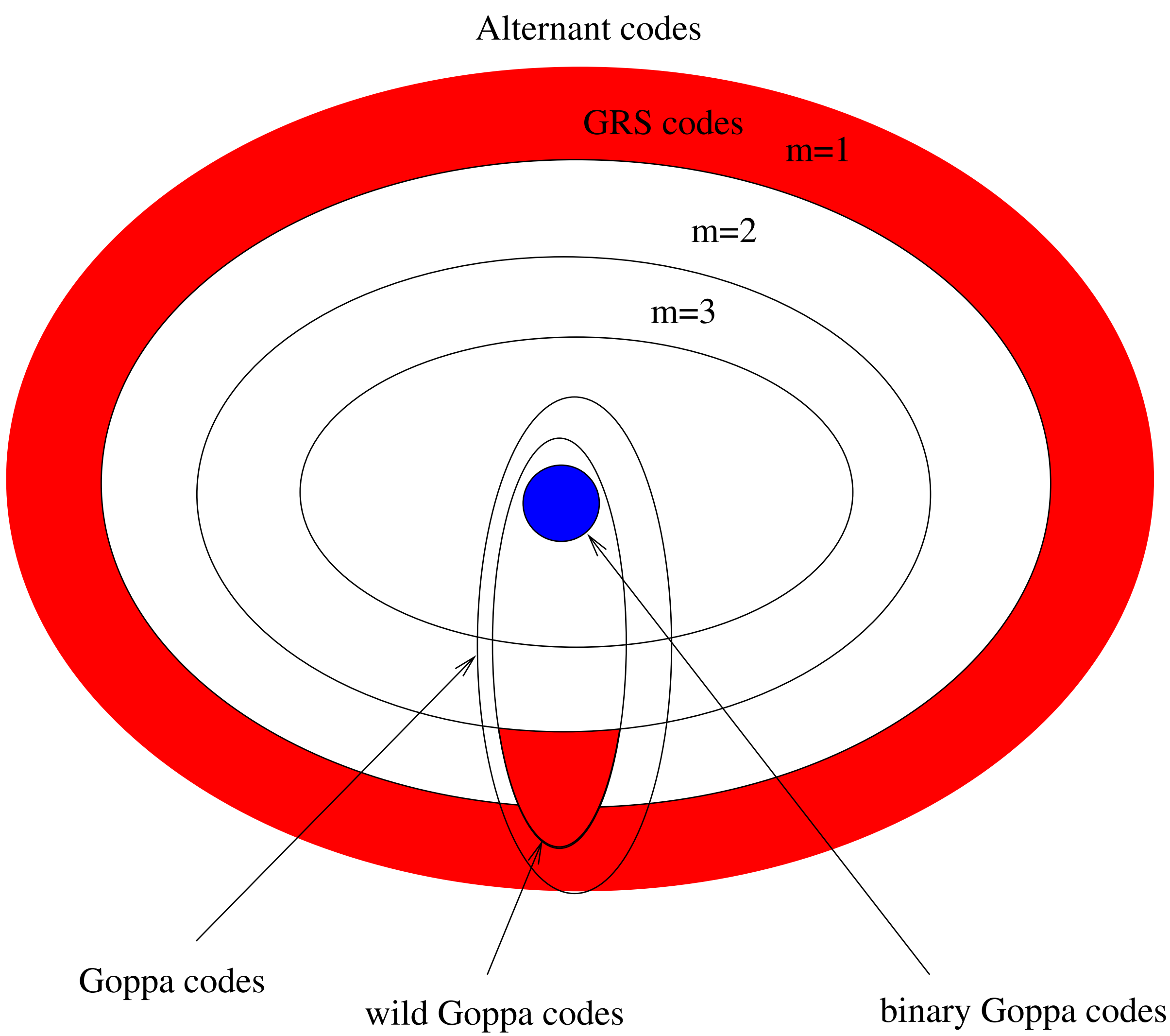
- $C_{k-1} = \{\alpha \boldsymbol{y}, \alpha \in \mathbb{F}_q\}$
- $\boldsymbol{y}$ known $\Rightarrow \boldsymbol{x}$ by solving a linear system.

The fundamental induction

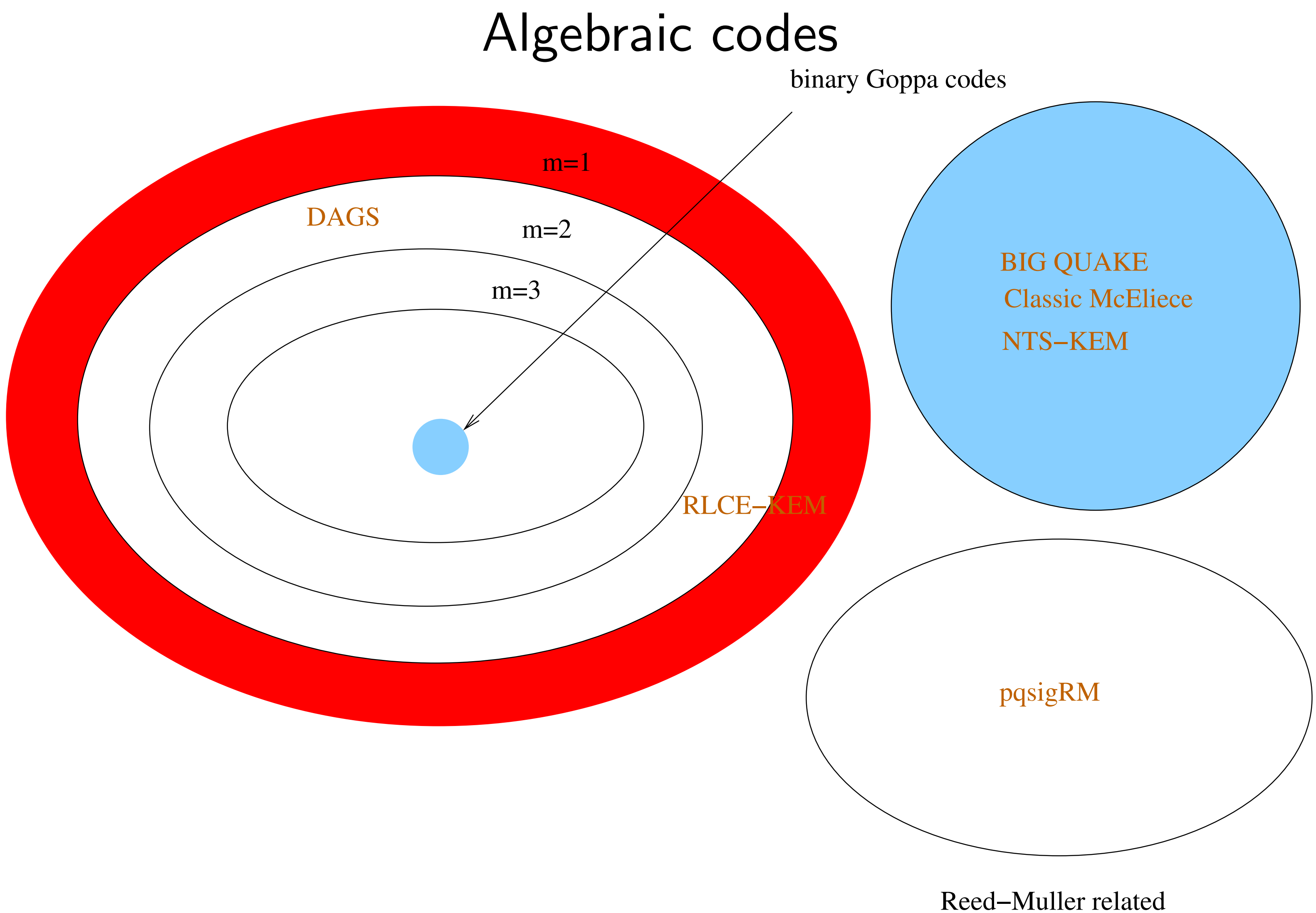
$$C_i \star C_{i-2} = C_{i-1} \star C_{i-1}$$

$$\begin{aligned} C_i \star C_{i-2} &= \mathbf{GRS}_{k-i}(\boldsymbol{x}, \boldsymbol{y}) \star \mathbf{GRS}_{k-i+2}(\boldsymbol{x}, \boldsymbol{y}) \\ &= \mathbf{GRS}_{2k-2i+1}(\boldsymbol{x}, \boldsymbol{y} \star \boldsymbol{y}) \\ &= \mathbf{GRS}_{k-i+1}(\boldsymbol{x}, \boldsymbol{y}) \star \mathbf{GRS}_{k-i+1}(\boldsymbol{x}, \boldsymbol{y}) \\ &= C_{i-1} \star C_{i-1} \end{aligned}$$

The picture


Alternant codes
GRS codes
m=1
m=2
m=3
Goppa codes
wild Goppa codes
binary Goppa codes

Code based NIST submissions in Hamming metric



4. Folding operation, the “Origami attack”

Origami attack

- Related to Gentry attack on NTRU-composite
- Applies to codes with a non trivial permutation group

For $\sigma \in S_n$,

$$\boldsymbol{c}^\sigma \stackrel{\text{def}}{=} (c_{\sigma(i)})_{i \in [\![1,n]\!]}$$

$$\mathcal{C}^\sigma \stackrel{\text{def}}{=} \{\boldsymbol{c}^\sigma : \boldsymbol{c} \in \mathcal{C}\}$$

σ is a permutation automorphism of $\mathcal{C}$ iff

$$\mathcal{C}^\sigma = \mathcal{C}$$

Examples

Parity-check matrix has a block form $\boldsymbol{H} = \begin{pmatrix} \boldsymbol{B}^{(11)} & \dots & \boldsymbol{B}^{(1n')} \\ \vdots & \boldsymbol{B}^{(ij)} & \vdots \\ \boldsymbol{B}^{(r'1)} & \dots & \boldsymbol{B}^{(r'n')} \end{pmatrix}$

with blocks of some size ℓ of the form

$$\boldsymbol{B}^{(ij)} = \begin{pmatrix} a_0 & a_1 & \cdots & a_{\ell-1} \\ a_{\ell-1} & a_0 & \cdots & a_{\ell-2} \\ \vdots & \ddots & \ddots & \vdots \\ a_1 & a_2 & \cdots & a_0 \end{pmatrix}$$

quasi-cyclic case $B^{(ij)}_{s,t} = a_{t-s \pmod{\ell}}$

$$\boldsymbol{B}^{(ij)} = \left(\begin{array}{cc|cc} a_0 & a_1 & a_2 & a_3 \\ a_1 & a_0 & a_3 & a_2 \\ \hline a_2 & a_3 & a_0 & a_1 \\ a_3 & a_2 & a_1 & a_0 \end{array}\right)$$

quasidyadic case $B^{(ij)}_{s,t} = a_{t \ominus s}$

Folding

- Folding $\boldsymbol{x} =$ w.r. to σ adding the coordinates in a same orbit of σ

$$\begin{aligned}
\sigma &= (123)(456)(678) \\
\boldsymbol{x} &= (\underbrace{x_1, x_2, x_3}_{\text{orbit}}, \dots, \underbrace{x_7, x_8, x_9}_{\text{orbit}}) \\
\overline{\boldsymbol{x}}^{\sigma} &= (x_1 + x_2 + x_3, \dots, x_7 + x_8 + x_8)
\end{aligned}$$

$$\overline{\mathcal{C}}^{\sigma} \stackrel{\text{def}}{=} \{\overline{\boldsymbol{c}}^{\sigma} : \boldsymbol{c} \in \mathcal{C}\}.$$

Why is this an interesting operation ?

Orbits of σ of size ℓ

- Code gets smaller

$$\begin{aligned} \mathcal{C} &= \text{code of length } n \text{ dim. } k \\ \to \overline{\mathcal{C}}^{\sigma} &= \text{code of length } n/\ell \text{ and dim. } \frac{k}{\ell} \end{aligned}$$

- Words do not increase their weight

$$|\boldsymbol{c}| = w \Rightarrow |\overline{\boldsymbol{c}}^{\sigma}| \leqslant w$$

Folding quasi-$*$ alternant codes/ Goppa codes

- [Faugère, Otmani, Perret, Portzamparc, T 2014] Folding the dual of a Q*-alternant or Q*-Goppa code $\Rightarrow$ dual of an alternant or a Goppa code
- [Barelli-Couvreur 2017] Folding a Q*-alternant or a Q*-Goppa code $\Rightarrow$ alternant or a Goppa code

Message attacks

$$\begin{cases} \boldsymbol{H}\boldsymbol{e}^\intercal & = & \boldsymbol{s}^\intercal \\ |\boldsymbol{e}| & \leqslant & t \end{cases}$$

$$\Rightarrow \begin{cases} \overline{\boldsymbol{H}}^\sigma (\overline{\boldsymbol{e}}^\sigma)^\intercal & = & (\overline{\boldsymbol{s}}^\sigma)^\intercal \\ |\overline{\boldsymbol{e}}^\sigma| & \leqslant & t \end{cases}$$

We recover $\overline{\boldsymbol{e}}^\sigma$ (say $= e_0$) and then solve the much easier problem

$$\begin{cases} \overline{\boldsymbol{H}}^\sigma \boldsymbol{e}^\intercal & = & \boldsymbol{s}^\intercal \\ |\boldsymbol{e}| & \leqslant & t \\ \overline{\boldsymbol{e}}^\sigma & = & \boldsymbol{e}_0 \end{cases}$$

5. Algebraic attacks

Alternant code $\mathbf{Alt}_r(\boldsymbol{x}, \boldsymbol{y})$ parity-check matrix $\boldsymbol{H}$ of the form

$$\boldsymbol{H} = \begin{bmatrix} y_1 & y_2 & \cdots & \cdots & y_n \\ y_1 x_1 & y_2 x_2 & \cdots & \cdots & y_n x_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & y_j x_j^i & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_1 x_1^{r-1} & y_2 x_2^{r-1} & \cdots & \cdots & y_n x_n^{r-1} \end{bmatrix}$$

Goppa code $\mathbf{Gop}(\boldsymbol{x}, \Gamma) = \mathbf{Alt}_{\deg \Gamma}(\boldsymbol{x}, \frac{1}{\Gamma(\boldsymbol{x})})$.

Algebraic attacks

$\boldsymbol{G} = (g_{ij})_{\substack{i \in [\![1,k]\!] \\ j \in [\![1,n]\!]}}$ generator matrix of $\mathcal{C} = \textbf{Alt}_r(\boldsymbol{x}, \boldsymbol{y})$.

Unknowns: $y_1, \ldots, y_n, \ x_1, \ldots, x_n$

$2n$ unknowns

Algebraic system

$$\begin{aligned} \boldsymbol{G}\boldsymbol{H}^\mathsf{T} = & \quad 0 \\ \Rightarrow \sum_{j=1}^{n} g_{ij} y_j x_j^a = & \quad 0 \qquad \forall (i,a) \in [\![1,k]\!] \times [\![0, r-1]\!] \end{aligned}$$

$k \cdot r$ equations

When was this successful ?

- [Faugère,Otmani,Perret,T 2010-2015] Q*-alternant of Q*-Goppa codes
- [Faugère,Perret,Portzamparc 2014] Wild Goppa codes for certain parameters

Rank Metric

Difficult problem in coding theory

Problem 2. [Decoding]
Input: n, r, t *integers,* $r < n$, *parity-check matrix* $\boldsymbol{H} \in {\mathbb{F}_{q^m}}^{r \times n}$*,*
syndrome $\boldsymbol{s} \in \mathbb{F}_q^r$
Question: $\exists$*?* $\boldsymbol{e}$ *such that*
(i) $\boldsymbol{He} = \boldsymbol{s}$*,*
(ii) $|\boldsymbol{e}| \leqslant t$
where $|\boldsymbol{e}|_R =$ *rank weight of* $\boldsymbol{e}$*.*

Randomized reduction to NP-complete problems.

Rank metric

- $(\beta_1 \dots \beta_m)$ basis of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$

$$\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{F}_{q^m}^n \to \textbf{Mat}(\boldsymbol{x}) = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix} \in \mathbb{F}_q^{m \times n}$$

where $x_j = \sum_{i=1}^{m} x_{ij}\beta_i$.

- Rank metric = viewing an element of $\mathbb{F}_{q^m}^n$ as an $m \times n$ matrix.

$$|\boldsymbol{x} - \boldsymbol{y}|_r \stackrel{\text{def}}{=} \textbf{Rank}\left(\textbf{Mat}(\boldsymbol{x}) - \textbf{Mat}(\boldsymbol{y})\right).$$

Complexity of the best known algorithms

- Algebraic attacks (MinRank)
- Combinatorial attacks $\tilde{O}\left(q^{t(k+1)-m}\right)$ when $m = n$.

LRPC codes

[Gaborit, Murat, Ruatta, Zémor 2013]

Definition 4. *An LRPC code over $\mathbb{F}_{q^m}$ of weight d is a code that admits an $(n-k)\times n$ parity-check matrix $\boldsymbol{H}$ with entries h_{ij} that span an $\mathbb{F}_q$ space of dimension d.*

$$|\boldsymbol{x}|_r = \dim\langle x_1,\dots,x_n\rangle_{\mathbb{F}_q}$$

$\Rightarrow$ all rows of $\boldsymbol{H}$ have weight $\leqslant d$.

- Correct t errors when $td \leqslant n-k$.

RankSign

Secret key $\boldsymbol{H}'$ where

$$\boldsymbol{H}' = [\boldsymbol{H}|\boldsymbol{R}]\,\boldsymbol{P}$$

with

$$\begin{aligned}
\boldsymbol{H} &= (n-k)\times n \text{ parity-check matrix of an LRPC code over } \mathbb{F}_{q^m}\\
\boldsymbol{R} &= \text{random } (n-k)\times t \text{ matrix over } \mathbb{F}_{q^m}\\
\boldsymbol{P} &= (n+t)\times(n+t) \text{ invertible matrix over } \mathbb{F}_q
\end{aligned}$$

- $\boldsymbol{P}$ isometry $|\boldsymbol{x}\boldsymbol{P}|_r = |\boldsymbol{x}|_r$.
- LRPC code of weight $d \Rightarrow$ codewords of weight $\leqslant d+t$ in the dual code.

Attack on RankSign

[Debris-Alazard, T 2018]

- Looking for low weight codewords in the dual code?

Attack on RankSign

[Debris-Alazard, T 2018]

- Looking for low weight codewords in the code itself
- Product trick

Getting rid of R

If there is a low weight codeword $\boldsymbol{c}_{\text{LRPC}}$ in $\mathcal{C}_{\text{LRPC}}$ $\Rightarrow$ low weight codeword $\boldsymbol{c}' = (\boldsymbol{c}_{\text{LRPC}}, \mathbf{0}_t)(\boldsymbol{P}^{-1})^{\intercal}$ in the public code of parity-check matrix $\boldsymbol{H}_{\text{pub}} = \boldsymbol{Q}\boldsymbol{H}' = \left[\boldsymbol{H}|\boldsymbol{R}\right]\boldsymbol{P}$

$$
\begin{aligned}
\boldsymbol{H}_{\text{pub}}\boldsymbol{c}'^{\intercal} &= \boldsymbol{H}_{\text{pub}}\boldsymbol{P}^{-1}(\boldsymbol{c}_{\text{LRPC}}, \mathbf{0}_t)^{\intercal} \\
&= \boldsymbol{Q}\left[\boldsymbol{H}|\boldsymbol{R}\right]\boldsymbol{P}\boldsymbol{P}^{-1}(\boldsymbol{c}_{\text{LRPC}}, \mathbf{0}_t)^{\intercal} \\
&= \boldsymbol{Q}\left[\boldsymbol{H}|\boldsymbol{R}\right](\boldsymbol{c}_{\text{LRPC}}, \mathbf{0}_t)^{\intercal} \\
&= \boldsymbol{Q}\boldsymbol{H}\boldsymbol{c}_{\text{LRPC}}{}^{\intercal} \quad (\ \boldsymbol{R} \in \mathbb{F}_{q^m}^{(n-k)\times t}\ ) \\
&= \mathbf{0} \quad (\boldsymbol{c}_{\text{LRPC}} \text{ belongs to the code of parity-check matrix } \boldsymbol{H})
\end{aligned}
$$

Product trick

F $\mathbb{F}_q$-space of dimension d generated by the entries of $\boldsymbol{H}$ parity-check of the $[n, k]$ LRPC code $\mathcal{C}_{\mathsf{LRPC}}$. U and V two subspaces of $\mathbb{F}_{q^m}$,

$$U \cdot V \stackrel{\text{def}}{=} \langle uv : u \in U,\ v \in V \rangle_{\mathbb{F}_q}.$$

Lemma 1. *It there exists an $\mathbb{F}_q$-subspace F' of $\mathbb{F}_{q^m}$ such that*

$$(n-k)\dim(F \cdot F') < n \cdot \dim F'.$$

Then there exist nonzero codewords in the LRPC code of weight $\leqslant \dim F'$.

Proof

A codeword $\boldsymbol{c}$ of the LRPC code satisfies

$$\forall i \in [\![1, n-k]\!] \quad \sum_{j=1}^{n} H_{i,j} c_j = 0. \tag{1}$$

If its entries are in F' then $\sum_{j=1}^{n} H_{i,j} c_j \in F \cdot F'$
unknowns coordinates c_{ij} of c_j in $F' = \langle f'_1, \dots, f'_{d'} \rangle_{\mathbb{F}_q}$:

$$c_j = \sum_{i \in [\![1, d']\!]} c_{ij} f'_i$$

$$\begin{aligned} \#\text{ equations} &= (n-k) \dim F \cdot F' \\ \#\text{unknowns} &= n \dim F' \end{aligned}$$

Consequence on RankSign

- Necessary condition for RankSign to work $n = (n-k)d$
- Problem: typically $\dim F \cdot F' = \dim F \dim F'$ and therefore

$$n \dim F' = n \cdot d' = (n-k)d \cdot d' = (n-k) \dim F \cdot F'$$

$$\begin{array}{rcl} F & = & \langle f_1, \dots, f_d \rangle_{\mathbb{F}_q} \\ F' & \stackrel{\text{def}}{=} & \langle f_1, f_2 \rangle_{\mathbb{F}_q} \\ F\dot{F}' & = & \langle x_i x_j : i \in [\![1,d]\!],\ j \in [\![1,2]\!] \rangle_{\mathbb{F}_q} \\ \dim F \cdot F' & = & 2d-1 < \dim F \dim F' \\ & \Rightarrow & \text{codewords in } \mathcal{C}_{\mathsf{LRPC}} \text{ of weight } 2 \end{array}$$

Consequence on LRPC in general ?

- No direct attack on LRPC codes without the additional condition $n = (n-k)d$

Conclusion

- Up to now all distinguishers of the public parity-check matrix / random matrix $\Rightarrow$ with the exception of high rate alternant/Goppa codes.
- [Faugère,Gauthier,Otmani,Perret,T 2011], [Márquez-Corbella, Pellikaan 2012], when r is sufficiently small

$$\dim\left(\textsf{\textbf{Alt}}_r(\boldsymbol{x},\boldsymbol{y})^{\perp} \star \textsf{\textbf{Alt}}_r(\boldsymbol{x},\boldsymbol{y})^{\perp}\right) \text{ unusually small}$$

The problem, when $\boldsymbol{x}$, $\boldsymbol{y} \in \mathbb{F}_{q^m}^n$

$$\begin{aligned} \textsf{\textbf{Alt}}_r(\boldsymbol{x},\boldsymbol{y}) &= \{(y_j p(x_j)) : \deg p < n-r\} \cap \mathbb{F}_q^n \\ \textsf{\textbf{Alt}}_r(\boldsymbol{x},\boldsymbol{y})^{\perp} &= \left\{\left(\textsf{\textbf{Tr}}_{\mathbb{F}_{q^m}\to\mathbb{F}_q}(y_j p(x_j))\right) : \deg p < r\right\} \end{aligned}$$

Other open problems

- improving algebraic attacks in the rank metric
- Polynomial time attacks on Reed-Muller codes ?
- other families of codes (MDPC,. . .)?

What about alternant/Goppa codes ?

We have

$$\begin{aligned}\mathbf{Alt}_r(\boldsymbol{x},\boldsymbol{y}) &= \mathbf{GRS}_r(\boldsymbol{x},\boldsymbol{y})^{\perp} \cap \mathbb{F}_q^n \\ &= \mathbf{GRS}_{n-r}(\boldsymbol{x},\boldsymbol{y}') \cap \mathbb{F}_q^n \\ \mathbf{Alt}_r(\boldsymbol{x},\boldsymbol{y})^2 &\subseteq \mathbf{Alt}_{2r-n+1}(\boldsymbol{x},\boldsymbol{y}')\end{aligned}$$

and

$$\dim \mathbf{Alt}_r(\boldsymbol{x},\boldsymbol{y}) \geqslant n - mr.$$

Fact 1. *To distinguish we need*

$$2r - n + 1 > 0 \quad \Longrightarrow \quad r \geqslant n/2,$$

however

$$m > 1 \quad \Longrightarrow n - mr \leqslant 0.$$

A miracle when $m = 2$ in the case of wild Goppa codes

Theorem 1. [Couvreur, Otmani, Tillich] *When* $\mathbf{Alt}_r(\boldsymbol{x}, \boldsymbol{y})$ *is a wild Goppa code (here* $r = (q-1)r'$*)*

$$\mathbf{Alt}_r(\boldsymbol{x}, \boldsymbol{y}) \quad \geqslant n - 2r + r'(r' - 2)$$

and for r close to $n/2$ we may have wild Goppa codes of small dimension such that

$$2r - n + 1 > 0$$

Shortening trick for other dimensions

A shortened alternant code is still an alternant code of the same degree r as the original alternant code.

- Leads to a distinguisher of wild Goppa codes when $m = 2$
- Leads to an attack of the McEliece scheme based on wild Goppa codes when $m = 2$. First time that there is an attack working in polynomial time on a McEliece scheme based on Goppa codes.