

Code-Based Cryptography

McEliece Cryptosystem

Code-Based Cryptography

1. Error-Correcting Codes and Cryptography
2. McEliece Cryptosystem
3. Message Attacks (ISD)
4. **Key Attacks**
5. Other Cryptographic Constructions Relying on Coding Theory

4. Key Attacks

1. Introduction
2. Support Splitting Algorithm
3. Distinguisher for GRS codes
4. Attack against subcodes of GRS codes
5. Error-Correcting Pairs
6. Attack against GRS codes
7. Attack against Reed-Muller codes
8. **Attack against Algebraic Geometry codes**
9. Goppa codes still resist

Algebraic Geometry (AG) codes

→ An **Algebraic Geometry (AG)** code is defined by a triplet

$$\left(\mathcal{X}, \mathcal{P}, \mathcal{E} \right)$$

Algebraic Geometry (AG) codes

→ An **Algebraic Geometry (AG)** code is defined by a triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

\mathcal{X} is an algebraic curve
of genus g over the finite
field \mathbb{F}_q

Algebraic Geometry (AG) codes

→ An **Algebraic Geometry (AG)** code is defined by a triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

\mathcal{X} is an algebraic curve of genus g over the finite field \mathbb{F}_q

$\mathcal{P} = (P_1, \dots, P_n)$
is an n -tuple of distinct \mathbb{F}_q -rational points of \mathcal{X}

Algebraic Geometry (AG) codes

→ An Algebraic Geometry (AG) code is defined by a triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

\mathcal{X} is an algebraic curve of genus g over the finite field \mathbb{F}_q

$\mathcal{P} = (P_1, \dots, P_n)$
is an n -tuple of distinct \mathbb{F}_q -rational points of \mathcal{X}

E is an \mathbb{F}_q -divisor of \mathcal{X} such that $P_i \notin \text{supp}(E)$

Algebraic Geometry (AG) codes

→ An Algebraic Geometry (AG) code is defined by a triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

\mathcal{X} is an algebraic curve of genus g over the finite field \mathbb{F}_q

$\mathcal{P} = (P_1, \dots, P_n)$
is an n -tuple of distinct \mathbb{F}_q -rational points of \mathcal{X}

E is an \mathbb{F}_q -divisor of \mathcal{X} such that $P_i \notin \text{supp}(E)$

Algebraic Geometry (AG) codes

The AG code associated to the triplet $(\mathcal{X}, \mathcal{P}, E)$ is

$$C_L(\mathcal{X}, \mathcal{P}, E) = \{\text{ev}_{\mathcal{P}}(f) = (f(P_1), \dots, f(P_n)) \mid f \in L(E)\}$$

Algebraic Geometry (AG) codes

AG codes are "almost" optimal codes

Let $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$. Then,

$$d(\mathcal{C}) \geq n - K(\mathcal{C}) + 1 - g$$

where g is the genus of \mathcal{X}

Algebraic Geometry (AG) codes

AG codes are "almost" optimal codes

Let $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$. Then,

$$d(\mathcal{C}) \geq n - K(\mathcal{C}) + 1 - g$$

where g is the genus of \mathcal{X}

The dual of an AG code is an AG code

$$\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E^\perp)$$

AG codes for the McEliece scheme

Algebraic-Geometry codes



H. Janwa and O. Moreno.

McEliece public crypto system using algebraic-geometric codes.
Designs, Codes and Cryptography, 1996.

Parameters	Key size	Security level
$[171, 109, 61]_{128}$	16 ko	2^{66}

AG codes for the McEliece scheme

Algebraic-Geometry codes



H. Janwa and O. Moreno.

McEliece public crypto system using algebraic-geometric codes.
Designs, Codes and Cryptography, 1996.

Parameters	Key size	Security level
$[171, 109, 61]_{128}$	16 ko	2^{66}



Attack against this proposal:



C. Faure and L. Minder.

Cryptanalysis of the McEliece cryptosystem over hyperelliptic codes.
Proceedings 11th Int. Workshop on Algebraic and Combinatorial Coding Theory, 2008.



A. Couvreur, I. Márquez-Corbella and R. Pellikaan.

A Polynomial Time Attack against Algebraic Geometry Code Based Public Key Cryptosystems.
IEEE Information Theory, ISIT 2014, 1446-1450, 2014.

GRS codes are AG codes

→ Consider the **(AG)** code defined by the triplet

$$\left(\mathcal{X}, \mathcal{P}, \mathcal{E} \right)$$

GRS codes are AG codes

→ Consider the **(AG)** code defined by the triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

Consider the projective
curve $\mathcal{X} = \mathbb{P}^1$ given by
 $z = 0$

GRS codes are AG codes

→ Consider the **(AG)** code defined by the triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

Consider the projective curve $\mathcal{X} = \mathbb{P}^1$ given by $z = 0$

$$\mathcal{P} = (P_1, \dots, P_n)$$

where $P_j = (a_j : 1)$ for all $j = 1, \dots, n$

GRS codes are AG codes

→ Consider the **(AG)** code defined by the triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

Consider the projective curve $\mathcal{X} = \mathbb{P}^1$ given by $z = 0$

$\mathcal{P} = (P_1, \dots, P_n)$
where $P_j = (a_j : 1)$ for all $j = 1, \dots, n$

$E = (k - 1)P_\infty$
with $P_\infty = (1 : 0)$

GRS codes are AG codes

→ Consider the **(AG)** code defined by the triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

Consider the projective curve $\mathcal{X} = \mathbb{P}^1$ given by $z = 0$

$$\mathcal{P} = (P_1, \dots, P_n)$$

where $P_j = (a_j : 1)$ for all $j = 1, \dots, n$

$$E = (k-1)P_\infty$$

with $P_\infty = (1 : 0)$

A basis for $L(E)$ is given by $\left\{ 1, \frac{x}{y}, \frac{x^2}{y^2}, \dots, \frac{x^{k-1}}{y^{k-1}} \right\}$

GRS codes are AG codes

→ Consider the (AG) code defined by the triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

Consider the projective curve $\mathcal{X} = \mathbb{P}^1$ given by $z = 0$

$$\mathcal{P} = (P_1, \dots, P_n)$$

where $P_j = (a_j : 1)$ for all $j = 1, \dots, n$

$$E = (k-1)P_\infty$$

with $P_\infty = (1 : 0)$

A basis for $L(E)$ is given by $\left\{ 1, \frac{x}{y}, \frac{x^2}{y^2}, \dots, \frac{x^{k-1}}{y^{k-1}} \right\}$

Generator matrix
for $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{k-1} & a_2^{k-1} & \dots & a_n^{k-1} \end{pmatrix}$$

GRS codes are AG codes

→ Consider the (AG) code defined by the triplet

$$(\mathcal{X}, \mathcal{P}, E)$$

Consider the projective curve $\mathcal{X} = \mathbb{P}^1$ given by $z = 0$

$$\mathcal{P} = (P_1, \dots, P_n)$$

where $P_j = (a_j : 1)$ for all $j = 1, \dots, n$

$$E = (k-1)P_\infty$$

with $P_\infty = (1 : 0)$

A basis for $L(E)$ is given by $\left\{ 1, \frac{x}{y}, \frac{x^2}{y^2}, \dots, \frac{x^{k-1}}{y^{k-1}} \right\}$

Generator matrix
for $\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$

$$\begin{pmatrix} 1 & 1 & \dots & 1 \\ a_1 & a_2 & \dots & a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{k-1} & a_2^{k-1} & \dots & a_n^{k-1} \end{pmatrix}$$

Generator matrix
for $\text{GRS}_k(\mathbf{a}, \mathbf{1})$

Filtration Attack for GRS codes - Retrieving an ECP

Suppose that we know:

$$\mathcal{C}_k = \text{GRS}_k(\mathbf{a}, \mathbf{b}) \quad \text{and} \quad \mathcal{C}_{k-1} = \text{GRS}_{k-1}(\mathbf{a}, \mathbf{b})$$

Proposition: Assume that $2k - 1 \leq n - 2$

$\mathcal{C}_{k-2} = \text{GRS}_{k-2}(\mathbf{a}, \mathbf{b})$ is the solution space of the following problem

$$\mathbf{c} \in \mathcal{C}_{k-1} \quad \text{and} \quad \mathbf{c} * \mathcal{C}_k \subseteq (\mathcal{C}_{k-1})^2$$

Filtration Attack for GRS codes - Retrieving an ECP

Suppose that we know:

$$\mathcal{C}_k = \text{GRS}_k(\mathbf{a}, \mathbf{b}) \quad \text{and} \quad \mathcal{C}_{k-1} = \text{GRS}_{k-1}(\mathbf{a}, \mathbf{b})$$

Proposition: Assume that $2k - 1 \leq n - 2$

$\mathcal{C}_{k-2} = \text{GRS}_{k-2}(\mathbf{a}, \mathbf{b})$ is the solution space of the following problem

$$\mathbf{c} \in \mathcal{C}_{k-1} \quad \text{and} \quad \mathbf{c} * \mathcal{C}_k \subseteq (\mathcal{C}_{k-1})^2$$

In this way we build a filtration

$$\text{GRS}_k(\mathbf{a}, \mathbf{b}) \supseteq \text{GRS}_{k-1}(\mathbf{a}, \mathbf{b}) \supseteq \text{GRS}_{k-2}(\mathbf{a}, \mathbf{b}) \supseteq \dots$$

Filtration Attack for GRS codes - Retrieving an ECP

Proposition 1:

Let $\mathcal{C} = \text{GRS}_k(\mathbf{c}, \mathbf{d}) \implies \mathcal{C}^\perp = \text{GRS}_{n-k}(\mathbf{c}, \mathbf{d}^\perp)$

Filtration Attack for GRS codes - Retrieving an ECP

Proposition 1:

$$\text{Let } \mathcal{C} = \text{GRS}_k(\mathbf{c}, \mathbf{d}) \implies \mathcal{C}^\perp = \text{GRS}_{n-k}(\mathbf{c}, \mathbf{d}^\perp)$$

Then, $\mathcal{A} = \text{GRS}_{t+1}(\mathbf{c}, \mathbf{1})$ and $\mathcal{B} = \text{GRS}_t(\mathbf{c}, \mathbf{d}^\perp)$
is a t -ECP for \mathcal{C} over \mathbb{F}_q

Filtration Attack for GRS codes - Retrieving an ECP

Proposition 1:

$$\text{Let } \mathcal{C} = \text{GRS}_k(\mathbf{c}, \mathbf{d}) \implies \mathcal{C}^\perp = \text{GRS}_{n-k}(\mathbf{c}, \mathbf{d}^\perp)$$

$$\text{Then, } \mathcal{A} = \text{GRS}_{t+1}(\mathbf{c}, \mathbf{1}) \quad \text{and} \quad \mathcal{B} = \text{GRS}_t(\mathbf{c}, \mathbf{d}^\perp)$$

is a t -ECP for \mathcal{C} over \mathbb{F}_q

Proposition 2: To compute a t -ECP for $\mathcal{C} = \text{GRS}_k(\mathbf{a}, \mathbf{b})$ it suffices to compute a code of type $\mathcal{B} = \text{GRS}_t(\mathbf{c}, \mathbf{d}^\perp)$

$$\text{If we know } \mathcal{C} = \text{GRS}_k(\mathbf{c}, \mathbf{d}) \quad \text{and} \quad \mathcal{B} = \text{GRS}_t(\mathbf{c}, \mathbf{d}^\perp)$$

Filtration Attack for GRS codes - Retrieving an ECP

Proposition 1:

$$\text{Let } \mathcal{C} = \text{GRS}_k(\mathbf{c}, \mathbf{d}) \implies \mathcal{C}^\perp = \text{GRS}_{n-k}(\mathbf{c}, \mathbf{d}^\perp)$$

$$\text{Then, } \mathcal{A} = \text{GRS}_{t+1}(\mathbf{c}, \mathbf{1}) \quad \text{and} \quad \mathcal{B} = \text{GRS}_t(\mathbf{c}, \mathbf{d}^\perp)$$

is a t -ECP for \mathcal{C} over \mathbb{F}_q

Proposition 2: To compute a t -ECP for $\mathcal{C} = \text{GRS}_k(\mathbf{a}, \mathbf{b})$ it suffices to compute a code of type $\mathcal{B} = \text{GRS}_t(\mathbf{c}, \mathbf{d}^\perp)$

$$\text{If we know } \mathcal{C} = \text{GRS}_k(\mathbf{c}, \mathbf{d}) \quad \text{and} \quad \mathcal{B} = \text{GRS}_t(\mathbf{c}, \mathbf{d}^\perp)$$

$$\text{Then, } \mathcal{A} = \text{GRS}_{t+1}(\mathbf{a}, \mathbf{1}) = (\mathcal{B} * \mathcal{C})^\perp$$

Filtration Attack for GRS codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $2k - 1 \leq n - 2$

Filtration Attack for GRS codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $2k - 1 \leq n - 2$

1. Determine the codes

$$\mathcal{C}_{\text{pub}}^\perp = \text{GRS}_{2t}(\mathbf{a}, \mathbf{b}^\perp) \quad \text{and} \quad \mathcal{S}_1(\mathcal{C}_{\text{pub}}^\perp) = \text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)$$

Filtration Attack for GRS codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $2k - 1 \leq n - 2$

1. Determine the codes

$$\mathcal{C}_{\text{pub}}^\perp = \text{GRS}_{2t}(\mathbf{a}, \mathbf{b}^\perp) \quad \text{and} \quad \mathcal{S}_1(\mathcal{C}_{\text{pub}}^\perp) = \text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)$$

2. Build the filtration:

$$\underbrace{\text{GRS}_{2t}(\mathbf{a}, \mathbf{b})}_{\mathcal{C}_{\text{pub}}^\perp} \supseteq \underbrace{\text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)}_{\mathcal{S}_1(\mathcal{C}_{\text{pub}}^\perp)} \supseteq \dots \supseteq \underbrace{\text{GRS}_t(\mathbf{a}', \sim \mathbf{b}^\perp)}_{\mathcal{B}}$$

Filtration Attack for GRS codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $2k - 1 \leq n - 2$

1. Determine the codes

$$\mathcal{C}_{\text{pub}}^\perp = \text{GRS}_{2t}(\mathbf{a}, \mathbf{b}^\perp) \quad \text{and} \quad \mathcal{S}_1(\mathcal{C}_{\text{pub}}^\perp) = \text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)$$

2. Build the filtration:

$$\underbrace{\text{GRS}_{2t}(\mathbf{a}, \mathbf{b})}_{\mathcal{C}_{\text{pub}}^\perp} \supseteq \underbrace{\text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)}_{\mathcal{S}_1(\mathcal{C}_{\text{pub}}^\perp)} \supseteq \dots \supseteq \underbrace{\text{GRS}_t(\mathbf{a}', \sim \mathbf{b}^\perp)}_{\mathcal{B}}$$

Generator matrix for \mathcal{C}_k $\leftarrow G = \begin{pmatrix} 1 & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{k2} & \dots & a_{kn} \end{pmatrix}$

Filtration Attack for GRS codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $2k - 1 \leq n - 2$

1. Determine the codes

$$\mathcal{C}_{\text{pub}}^\perp = \text{GRS}_{2t}(\mathbf{a}, \mathbf{b}^\perp) \quad \text{and} \quad S_1(\mathcal{C}_{\text{pub}}^\perp) = \text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)$$

2. Build the filtration:

$$\underbrace{\text{GRS}_{2t}(\mathbf{a}, \mathbf{b})}_{\mathcal{C}_{\text{pub}}^\perp} \supseteq \underbrace{\text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)}_{S_1(\mathcal{C}_{\text{pub}}^\perp)} \supseteq \dots \supseteq \underbrace{\text{GRS}_t(\mathbf{a}', \sim \mathbf{b}^\perp)}_{\mathcal{B}}$$

Generator matrix for \mathcal{C}_k $\leftarrow G = \begin{pmatrix} 1 & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{k2} & \dots & a_{kn} \end{pmatrix}$ Generator matrix for $S_1(\mathcal{C}_k)$

Filtration Attack for GRS codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $2k - 1 \leq n - 2$

1. Determine the codes

$$\mathcal{C}_{\text{pub}}^\perp = \text{GRS}_{2t}(\mathbf{a}, \mathbf{b}^\perp) \quad \text{and} \quad \mathcal{S}_1(\mathcal{C}_{\text{pub}}^\perp) = \text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)$$

2. Build the filtration:

$$\underbrace{\text{GRS}_{2t}(\mathbf{a}, \mathbf{b})}_{\mathcal{C}_{\text{pub}}^\perp} \supseteq \underbrace{\text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)}_{\mathcal{S}_1(\mathcal{C}_{\text{pub}}^\perp)} \supseteq \dots \supseteq \underbrace{\text{GRS}_t(\mathbf{a}', \sim \mathbf{b}^\perp)}_{\mathcal{B}}$$

3. Return $(\mathcal{A}, \mathcal{B})$ which is an *ECP* for $\mathcal{S}_1(\mathcal{C})$ where: $\mathcal{A} = (\mathcal{B} * \mathcal{S}_1(\mathcal{C}))^\perp$

Filtration Attack for GRS codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \text{GRS}_k(\mathbf{a}, \mathbf{b}) \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $2k - 1 \leq n - 2$

1. Determine the codes

$$\mathcal{C}_{\text{pub}}^\perp = \text{GRS}_{2t}(\mathbf{a}, \mathbf{b}^\perp) \quad \text{and} \quad \mathcal{S}_1(\mathcal{C}_{\text{pub}}^\perp) = \text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)$$

2. Build the filtration:

$$\underbrace{\text{GRS}_{2t}(\mathbf{a}, \mathbf{b})}_{\mathcal{C}_{\text{pub}}^\perp} \supseteq \underbrace{\text{GRS}_{2t-1}(\mathbf{a}', \sim \mathbf{b}^\perp)}_{\mathcal{S}_1(\mathcal{C}_{\text{pub}}^\perp)} \supseteq \dots \supseteq \underbrace{\text{GRS}_t(\mathbf{a}', \sim \mathbf{b}^\perp)}_{\mathcal{B}}$$

3. Return $(\mathcal{A}, \mathcal{B})$ which is an *ECP* for $\mathcal{S}_1(\mathcal{C})$ where: $\mathcal{A} = (\mathcal{B} * \mathcal{S}_1(\mathcal{C}))^\perp$
4. **Note that:** Correcting an error in the first position is not a difficult problem.

Filtration Attack for AG codes - Retrieving an ECP

Suppose that we know:

$$\mathcal{C}_0 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E) \quad \text{and} \quad \mathcal{C}_1 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - P)$$

Proposition: Assume that $\frac{n}{2} - 2 \geq \deg(E)$

$\mathcal{C}_2 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - 2P)$ is the solution space of the following problem

$$\mathbf{c} \in \mathcal{C}_1 \quad \text{and} \quad \mathbf{c} * \mathcal{C}_0 \subseteq (\mathcal{C}_1)^2$$

Filtration Attack for AG codes - Retrieving an ECP

Suppose that we know:

$$C_0 = C_L(\mathcal{X}, \mathcal{P}, E) \quad \text{and} \quad C_1 = C_L(\mathcal{X}, \mathcal{P}, E - P)$$

Proposition: Assume that $\frac{n}{2} - 2 \geq \deg(E)$

$C_2 = C_L(\mathcal{X}, \mathcal{P}, E - 2P)$ is the solution space of the following problem

$$\mathbf{c} \in C_1 \quad \text{and} \quad \mathbf{c} * C_0 \subseteq (C_1)^2$$

In this way we build a filtration

$$\underbrace{C_L(\mathcal{X}, \mathcal{P}, E)}_{C_0} \supseteq \underbrace{C_L(\mathcal{X}, \mathcal{P}, E - P)}_{C_1} \supseteq \underbrace{C_L(\mathcal{X}, \mathcal{P}, E - 2P)}_{C_2} \supseteq \dots$$

Filtration Attack for AG codes - Retrieving an ECP

Proposition 1: [Pellikaan 1992]

Let $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$

Filtration Attack for AG codes - Retrieving an ECP

Proposition 1: [Pellikaan 1992]

Let $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$

Then, $\mathcal{A} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, F)$ and $\mathcal{B} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ with
 $\deg(E) > \deg(F) = t + g$ is a t -ECP for \mathcal{C}

Filtration Attack for AG codes - Retrieving an ECP

Proposition 1: [Pellikaan 1992]

Let $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$

Then, $\mathcal{A} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, F)$ and $\mathcal{B} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ with
 $\deg(E) > \deg(F) = t + g$ is a t -ECP for \mathcal{C}

Proposition 2: To compute a t -ECP for $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ it suffices to compute a code of type $\mathcal{B} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t + g)P)$

If we know

$\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$ and $\mathcal{B} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t + g)P)$

Filtration Attack for AG codes - Retrieving an ECP

Proposition 1: [Pellikaan 1992]

Let $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$

Then, $\mathcal{A} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, F)$ and $\mathcal{B} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - F)$ with
 $\deg(E) > \deg(F) = t + g$ is a t -ECP for \mathcal{C}

Proposition 2: To compute a t -ECP for $\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)$ it suffices to compute a code of type $\mathcal{B} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t + g)P)$

If we know

$\mathcal{C} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp$ and $\mathcal{B} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t + g)P)$

Let, $\mathcal{A} = (\mathcal{B} * \mathcal{C})^\perp$. Then the pair $(\mathcal{A}, \mathcal{B})$ is a t -ECP for \mathcal{C}

Filtration Attack for AG codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - g - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $\frac{n}{2} - 1 \geq \deg(E)$

Filtration Attack for AG codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - g - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $\frac{n}{2} - 1 \geq \deg(E)$

1. Determine the codes

$$\mathcal{C}_0 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E) = \mathcal{C}_{\text{pub}}^\perp \quad \text{and} \quad \mathcal{C}_1 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - P_1)$$

Note that \mathcal{C}_1 is the set of codewords of \mathcal{C}_0 which are zero at position P_1

Filtration Attack for AG codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - g - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $\frac{n}{2} - 1 \geq \deg(E)$

1. Determine the codes

$$\mathcal{C}_0 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E) = \mathcal{C}_{\text{pub}}^\perp \quad \text{and} \quad \mathcal{C}_1 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - P_1)$$

Note that \mathcal{C}_1 is the set of codewords of \mathcal{C}_0 which are zero at position P_1

Generator matrix
for \mathcal{C}_0 $\leftarrow G = \begin{pmatrix} 1 & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{k2} & \dots & a_{kn} \end{pmatrix}$

Filtration Attack for AG codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } C_{\text{pub}} = C_L(\mathcal{X}, \mathcal{P}, E)^\perp \\ \text{and } t = \left\lfloor \frac{d(C) - g - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $\frac{n}{2} - 1 \geq \deg(E)$

1. Determine the codes

$$C_0 = C_L(\mathcal{X}, \mathcal{P}, E) = C_{\text{pub}}^\perp \quad \text{and} \quad C_1 = C_L(\mathcal{X}, \mathcal{P}, E - P_1)$$

Note that C_1 is the set of codewords of C_0 which are zero at position P_1

Generator matrix for C_0 $\leftarrow G = \begin{pmatrix} 1 & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{k2} & \dots & a_{kn} \end{pmatrix}$ Generator matrix for C_1

Filtration Attack for AG codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - g - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $\frac{n}{2} - 1 \geq \deg(E)$

1. Determine the codes

$$\mathcal{C}_0 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E) = \mathcal{C}_{\text{pub}}^\perp \quad \text{and} \quad \mathcal{C}_1 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - P_1)$$

Note that \mathcal{C}_1 is the set of codewords of \mathcal{C}_0 which are zero at position P_1

2. Build the filtration:

$$\underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)}_{\mathcal{C}_{\text{pub}}^\perp = \mathcal{C}_0} \supseteq \underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - P_1)}_{\mathcal{C}_1} \supseteq \dots \supseteq \underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t+g)P_1)}_{\mathcal{C}_{t+g}}$$

Filtration Attack for AG codes - Retrieving an ECP

Public Key: $\mathcal{K}_{\text{pub}} = \left\{ \begin{array}{l} \text{a generator matrix of } \mathcal{C}_{\text{pub}} = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)^\perp \\ \text{and } t = \left\lfloor \frac{d(\mathcal{C}) - g - 1}{2} \right\rfloor \end{array} \right.$

The Algorithm: Assume that $\frac{n}{2} - 1 \geq \deg(E)$

1. Determine the codes

$$\mathcal{C}_0 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E) = \mathcal{C}_{\text{pub}}^\perp \quad \text{and} \quad \mathcal{C}_1 = \mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - P_1)$$

Note that \mathcal{C}_1 is the set of codewords of \mathcal{C}_0 which are zero at position P_1

2. Build the filtration:

$$\underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E)}_{\mathcal{C}_{\text{pub}}^\perp = \mathcal{C}_0} \supseteq \underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - P_1)}_{\mathcal{C}_1} \supseteq \dots \supseteq \underbrace{\mathcal{C}_L(\mathcal{X}, \mathcal{P}, E - (t+g)P_1)}_{\mathcal{C}_{t+g}}$$

3. Return $(\mathcal{A}, \mathcal{B})$ which is an *ECP* for \mathcal{C} where:

$$\mathcal{B} = \mathcal{C}_{t+g} \quad \text{and} \quad \mathcal{A} = (\mathcal{B} * \mathcal{C})^\perp$$

4. Key Attacks

1. Introduction
2. Support Splitting Algorithm
3. Distinguisher for GRS codes
4. Attack against subcodes of GRS codes
5. Error-Correcting Pairs
6. Attack against GRS codes
7. Attack against Reed-Muller codes
8. Attack against Algebraic Geometry codes
9. **Goppa codes still resist**