24th Conference on ACA
Santiago de Compostela
Session on CACTC

Computer Algebra Tales on Goppa Codes and McEliece Cryptography

N. Sayols & S. Xambó

UPC

19/6/2018
Ingredients of a McEliece cryptosystem (McECS)
The PyECC CAS
Construction of McECS
Security analysis and the post-quantum scenario
Code samples
Conclusions, discussion and future outlook
- $F = F_q$, a finite field of cardinal $q$ (*base field*). The most important case will be $F = \mathbb{Z}_2$.

- $k$ a positive integer. The vectors of $F^k$ are called *information vectors*, or *messages*.

- $n > k$ an integer. The vectors of $F^n$ are called *transmission vectors*. If $x \in F^n$, we let $|x|$ denote the number of non-zero components of $x$ and we say that it is the *weight* of $x$.

**Notations.** $F(r, s)$ denotes the space of matrices of type $r \times s$ with entries in $F$ and $F(r) = F(r, r)$. 
A receiving user needs the following data:

- $G \in F(k, n)$ such that $\text{rank}(G) = k$;
- $S \in F(k)$ invertible and chosen at random;
- $P \in F(n)$ a random permutation matrix;
- $t$, a positive integer; and
- $g : X \rightarrow F^k$, $X \subseteq F^n$, such that for any $u \in F^k$ and all $e \in F^n$ with $|e| \leq t$,

$$x = uG + e \in X \quad \text{and} \quad g(x) = u. \quad (1)$$

The map $g$ is called a $t$-error-correcting $G$-decoder, or simply decoder, and the vectors of $X$ are said to be $g$-decodable.

- **Private key:** $\{G, S, P\}$
- **Public key:** $\{G', t\}$, where $G' = SGP$. 
Encryption protocol

The protocol that a user has to follow to encrypt and send a message \( u \) to the user whose public key is \( \{ G', t \} \) consists of two steps:

- Random generation of a transmission vector \( e \) of weight \( t \);
- Sending the vector \( x = uG' + e = uSGP + e \) to that user.
Decryption protocol

Consists of four steps that only use private data of the receiver and the vector $x$ sent by the emitter:

- Set $y = xP^{-1}$, so that $y = (uS)G + eP^{-1}$.
- Set $x' = g(y)$. Since $P$ is a permutation matrix,
  \[ |eP^{-1}| = |e| = t, \]
  and hence $x'$ is well defined, as $g$ corrects $t$ errors. The result is $x' = (uS)G$, which says that $x'$ is the linear combination of the rows of $G$ with coefficients $u' = uS$.
- Since $G$ has rank $k$, $u'$ is uniquely determined by $x'$ and can be obtained by solving the system of linear equations $x' = u'G$, where $u'$ is the unknown vector.
- Let $u = u'S^{-1}$, which agrees with the message sent by the emitter.
```python
### pfactor: factorizing polynomials

```from PyECC.Fq import *
from PyECC.Power_Series import *
from PyECC.pfactorFunctions import *

...

Theorem: Let $K$ be a finite field of cardinal $q$, $f \in K[X]$, and $n = \text{degree}(f)$. Then $f$ is irreducible iff
1) $f$ divides $X^{(q^n)}$, and
2) $\gcd(X^{(q^n)(n/p)})$, $f = 1$ for all prime divisors $p$ of $n$.

See vzGathen-Gerhard-2003, pp 396-397.

```
Let $\alpha \in \mathbb{F}_2$ and assume that $\alpha^3 = \alpha + 1$. Consider the matrix

$$
H = \begin{pmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6
\end{pmatrix}
$$

and let $C$ be the alternant binary code associated to $H$. Let us see that $C \equiv [7, 3, 4]$, so that $d = 4 > 3 = r + 1$.

First the minimum distance $d$ of $C$ is $\geq 4$, as any three columns of $H$ are linearly independent over $\mathbb{F}_2$. On the other hand, the first three columns and the column of $\alpha^5$ are linearly dependent, for $\alpha^5 = \alpha^2 + \alpha + 1$, and so $d = 4$. Finally the dimension of $C$ is $3$, because it has a control matrix of rank $4$ over $\mathbb{F}_2$, as the the is shown by next CC script.

In [5]:

```python
## Computing the dimension using the blow and prune functions
from PyECC.CC import *

n = 7; r = 2
K = 2n(2);

[F,a] = extension(K,[1,0,1,1],'a','F')

H = create matrix(F,[[1,1,1,1,1,1,1],[1,a,a**2,a**3,a**4,a**5,a**6]])
show(blow(H,K))
show(prune (blow(H,K)))

[[0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0]
 [1 1 1 1 1 1 1]
 [0 0 1 0 1 1 1]
 [0 1 0 1 1 1 0]
 [1 0 0 1 0 1 1]] :: Matrix[ZZ]
```
A = Zn(17)
a = 3 >> A
--> 3 :: Z17

order(a)
--> 16

# Powers of a, seen as integers
[lift(a**j) for j in range(17)]
--> [1, 3, 9, 10, 13, 5, 15, 11, 16, 14, 8, 7, 4, 12, 2, 6, 1]

# Order of 3 mod 17
order(3, 17)
--> 16

# Powers of 3 mod 17
[3**j % 17 for j in range(17)]
--> [1, 3, 9, 10, 13, 5, 15, 11, 16, 14, 8, 7, 4, 12, 2, 6, 1]
To create the polynomial ring $P = A[X]$: 

$[P, X] = \text{polynomial\_ring}(A)$

To get an irreducible monic polynomial $f \in P$ of degree $t$: 

$f = \text{get\_irreducible\_polynomial}(P, t)$

**Gauss formula** for the number $I_q(t)$ of monic irreducible polynomials of degree $t$ over $F_q$: 

$$I_q(t) = \frac{1}{t} \sum_{d|t} \mu(t/d) q^d = \frac{q^t}{t} + \cdots$$

It follows that the probability of getting an irreducible polynomial out of all monic polynomial of degree $t$ is: 

$$\frac{I_q(t)}{q^t} = \frac{1}{t} + \cdots.$$
# Creating Z2
\[ K = \mathbb{Z}_n(2) \]

# Creating F = F8 as \( K[X]/(f=X^3+X+1) \), \( a = X \mod f \)
\[ [F,a] = \text{extension}(K,[1,0,1,1],'a','F') \]

# In general (A ring)
\[ [B,x] = \text{extension}(A,[1,a_1,\ldots,ar],'x','B') \]
# creates the ring
\[ B = A[X]/(f = X^r + a_1X^{r-1} + \cdots + a_{r-1}X + a_r), \]
If \( f = X^r + a_1X^{r-1} + \cdots + a_{r-1}X + a_r \in A[X] = P \), we also can use the following syntax:

\[ [P,X] = \text{polynomial\_ring}(A,'X') \]
\[ f = X**r + a_1*X**(r-1) + \cdots + a_{r-1}*X + a_r \]
\[ [B,x] = \text{extension}(A,f,'x','B') \]
# Creation of a length n vector
# with coefficients in the ring A
x = vector(A,n)

# creation of a matrix in A(k,n), A a ring
M = matrix(A,k,n)

# Assignment of values
x[j] = a
M[i,j] = a
The function `scramble_matrix(A,k)` creates a $S \in A(k)$ with $\det(A) = 1$ and which is random under these conditions.

Note that the function `rd(A)` returns an element of $A$ selected at random.

```python
def scramble_matrix(A,k):
    U = matrix(A,k,k)
    L = matrix(A,k,k)
    for i in range(k):
        U[i,i] = L[i,i] = 1
        for j in range(i+1,k):
            U[i,j] = rd(A)
            L[j,i] = rd(A)
    return L*U
```
The function `permutation_matrix(n)` creates a random permutation matrix of order $n$.

```python
def permutation_matrix(n):
    N = list(range(n))
    p = rd_choice(N, n)
    P = matrix(ZZ(), n, n)
    for j in range(n):
        P[j, p[j]] = 1
    return P
```
- $F = F_q$, $q = 2$ (many constructions work also for $q > 2$).
- $\bar{F} = F_{q^m}$, $m$ a positive integer. If $\beta \in \bar{F}$, $[\beta]$ will denote the column vector of its components with respect to a basis of $\bar{F}/F$.
- $\alpha = \alpha_1, \ldots, \alpha_n \in \bar{F}$ distinct elements, so that $n \leq q^m$.
- $p \in \bar{F}[X]$ a polynomial of degree $r > 0$ such that $p(\alpha_j) \neq 0$ ($j = 1, \ldots, n$).
- Set $h_j = 1/p(\alpha_j)$ ($j = 1, \ldots, n$) and
  
  $$\bar{H} = \begin{pmatrix}
  h_1 & \cdots & h_n \\
  h_1\alpha_1 & \cdots & h_n\alpha_n \\
  \vdots & & \vdots \\
  h_1\alpha_1^{r-1} & \cdots & h_n\alpha_n^{r-1}
  \end{pmatrix} \in \bar{F}(r, n).$$
Let $H \in F(r', n)$ be the result of replacing each entry $\beta$ of $\tilde{H}$ by $[\beta]$ (this yields a matrix $[H] \in F(mr, n)$), followed by deleting from $[H]$ any row that is in the span of the previous ones. Note that $r' \leq mr$. It also holds that $r \leq r'$, as the $\langle H \rangle_{\bar{F}} = \langle \tilde{H} \rangle_{\bar{F}}$.

Let $\Gamma = \Gamma(p, \alpha) = \{x \in F^n : xH^T = 0\}$. It is a code of type $[n, k = n - r']$. This code is called the classical Goppa code associated to $p$ and $\alpha$.

We have $n - mr \leq k \leq n - r$.

Fact: If $G \in F(k, n)$ is a generating matrix of $\Gamma$, there is $G$-decoder that corrects $r/2$ errors in general, and $r$ errors in the binary case provided $p$ has no multiple roots in $\bar{F}$. See, for example, [2, P.4.7]
Let $\alpha$ be the set of elements of $\bar{F}$. Hence $n = 2^m$.

Let $p \in \bar{F}[X]$ be a monic irreducible of degree $t > 1$. Then $p$ has no roots in $\bar{F}$ and so a generating matrix $G$ of $\Gamma(p, \alpha)$ has a decoder $g$ that corrects $t$ errors.

This ends the theoretical construction of a McECS with the following parameters:

- $n = 2^m$, where $m$ is any positive integer, and $p$ is monic irreducible of degree $t$.
- $\bar{H} \in \bar{F}(t, n)$ and $G \in F(k, n)$, where $k = n - \text{rank}(H)$ ($n - tm \leq k \leq n - t$).
- Original example: $m = 10$, $n = 1024$, $t = 50$, $k = 524$ (in this case $k = n - tm$, the minimum possible given $m$ and $t$).
Binary work factor (log scale)

Horizontal axis $R = k/n$, WF curves for $n = 2^j$, $j = 10, \ldots, 13$. 

- $n = 8192$
- $n = 4096$
- $n = 2048$
- $n = 1024$
```python
# Computing the dimension using the blow and prune functions
from CC import *

n = 7; r = 2
K = Zn(2);

[F,a] = extension(K,[1,0,1,1], 'a', 'F')

H = create_matrix(F,[[1,1,1,1,1,1,1], [1,a,a**2,a**3,a**4,a**5,a**6]])
show(blow(H,K))
show(prune(blow(H,K)))

[[ 0  0  0  0  0  0  0  0 ]
 [ 0  0  0  0  0  0  0  0 ]
 [ 1  1  1  1  1  1  1  1 ]
 [ 0  0  1  0  1  1  1  1 ]
 [ 0  1  0  1  1  1  0  0 ]
 [ 1  0  0  1  0  1  1  1 ]] :: Matrix[Z2]

[[ 1  1  1  1  1  1  1  1 ]
 [ 0  0  1  0  1  1  1  1 ]
 [ 0  1  0  1  1  1  0  0 ]
 [ 1  0  0  1  0  1  1  1 ]] :: Matrix[Z2]
```
F5 = Zn(5)
# Creation of F25, with generator x
[F25,x] = extension(F5,[1,0,-2],'x','F25')
# Creation of the polynomial ring F25[T]
[A,T] = polynomial_ring(F25,'T')
g = T**6 + T**3 + T + 1
a = Set(F25)[1:] # The non-zero elements of F25
a = [t for t in a if evaluate(g,t)!=0]
C = Goppa(g,a)
# generate a random error pattern of weight 3
e = rd_error_vector(Z5,n,3)
>> e = [0,1,0,0,0,3,0,4,0,0,0,0,0,0,0,0,0,0,0]
# Use the PGZ decoder for C
PGZ(e,C)
>>PGZ: Error positions [1,5,7], error values [1,3,4]
[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0] :: Vector[Z5]
• Code low level routines in cyton.
• Extend the package to include convolution codes.
• Second edition of Block Error-Correcting codes, based on PyECC
• Include PyECC in Sage?

More details: [5]
References (1)


References (2)

   First PQC Standardization Conference organized by the NIST Computer Security Resource Center.

Thanks, ...

Why are we doing that ...
Note that we cannot avoid higher cardinals because (high degree) extensions of the base field will play a crucial role.
- Purely Python.
- Hierarchy of classes driven by several function definition files.
- The function files are grouped in two components: Low level Modular arithmetic utilities (in red) and high level interface utilities (in blue).
- The classes are grouped in two subsets: Element classes (in green) and algebraic structures (in orange).
- The arrows correspond to interdependencies.
- For details, see PyECC