Non commutative Goppa codes and their use in code based cryptography[1]

J. Gómez-Torrecillas[†], F. J. Lobillo[‡] and G. Navarro[§]

[†]Department of Algebra and IMAG, University of Granada

[‡]Department of Algebra and CITIC, University of Granada

[§]Department of Computer Sciences and AI, and CITIC, University of Granada

Instituto de Matemáticas Universidad de Granada

UNIVERSIDAD DE GRANADA

CITIC-UGR

Coding Theory and Cryptography (CTC2022), July 14th, 2022

[1]Research funded by (PID2019-110525GB-I00 / AEI / 10.13039/501100011033).

Index

Ore polynomials.

Let σ be an automorphism of finite order μ of a field L. An additive map $\partial : L \to L$ is called a σ-derivation if it satisfies $\partial(ab) = \sigma(a)\partial(b) + \partial(a)b$ for all $a, b \in L$. By $R = L[x; \sigma, \partial]$ we denote the ring of Ore polynomials built from (σ, ∂). The product is defined by the rule

$$xa = \sigma(a)x + \partial(a).$$

R is a left and right Euclidean domain. The left division algorithm computes, given $f, d \in R$ with $d \neq 0$, two Ore polynomials $q, r \in R$ such that $f = qd + r$ with $\deg r < \deg d$, where $\deg$ denotes the degree (in x) function.
Left and Right Extended Euclidean Algorithms (LEEA and REEA) can be described in the usual form.

Skew differential Goppa codes.

Definition

Let $F \subseteq L$ be a field extension. Let $g \in R = L[x;\sigma,\partial]$ be a nonzero invariant polynomial, i.e. $Rg = gR$. Let $\alpha_0, \ldots, \alpha_{n-1} \in L$ be different P-independent elements, i.e.

$$\deg\left[\{x-\alpha_i \mid 0 \le i \le n-1\}\right]_\ell = n. \tag{1}$$

Assume $(x-\alpha_i, g)_r = 1$ for all $0 \le i \le n-1$ and let $h_i \in R$ such that $\deg(h_i) < \deg(g)$ and

$$(x-\alpha_i)h_i - 1 \in Rg. \tag{2}$$

Let also $\eta_0, \ldots, \eta_{n-1} \in L^*$. A (generalized) skew differential Goppa code is

$$\mathcal{C} = \left\{(c_0, \ldots, c_{n-1}) \in F^n \mid \textstyle\sum_{i=0}^{n-1} h_i \eta_i c_i \in Rg\right\}. \tag{3}$$

We say that $\{\alpha_0, \ldots, \alpha_{n-1}\}$ are the positional points, g is the (skew differential) Goppa polynomial and $h_0, \ldots, h_{n-1}$ are the parity check polynomials. If $\partial = 0$, we just call it a (generalized) skew Goppa code.

Syndrome, locator and evaluator polynomials

Assume $c \in \mathcal{C}$ is transmitted and $r \in F^n$ is received. Therefore $r = c + e$ for some $e = \sum_{j=1}^{\nu} e_j \varepsilon_{k_j}$ with $e_j \neq 0$ for $1 \leq j \leq \nu$. The *syndrome polynomial* is defined and computed as

$$s = \sum_{i=0}^{n-1} h_i \eta_i r_i.$$

By (3) it follows that

$$s - \sum_{j=1}^{\nu} h_{k_j} \eta_{k_j} e_j = \sum_{i=0}^{n-1} h_i \eta_i c_i \in Rg = gR. \tag{4}$$

We define the (non-commutative) *error locator polynomial* as

$$\lambda = \left[\{ x - \alpha_{k_j} \mid 1 \leq j \leq \nu \} \right]_\ell \in R.$$

Then $\deg(\lambda) = \nu$ and, for each $1 \leq j \leq \nu$, there exists $\rho_{k_j} \in R$ such that $\deg(\rho_{k_j}) = \nu - 1$ and

$$\lambda = \rho_{k_j}(x - \alpha_{k_j}). \tag{5}$$

The *error evaluator polynomial* is defined as

$$\omega = \sum_{j=1}^{\nu} \rho_{k_j} \eta_{k_j} e_j.$$

It follows that $\deg(\omega) < \nu$.

Key equation

Theorem

The error locator λ and the error evaluator ω polynomials satisfy the non-commutative key equation

$$\omega = \kappa g + \lambda s, \tag{6}$$

for some $\kappa \in R$. Assume that $\nu \leq t = \left\lfloor \frac{\deg g}{2} \right\rfloor$. Let u_l, v_l and r_l be the Bezout coefficients returned by the LEEA with input g and s, where l is the index determined by the conditions $\deg r_{l-1} \geq t$ and $\deg r_l < t$. Then there exists $h \in R$ such that $\kappa = hu_l$, $\lambda = hv_l$ and $\omega = hr_l$.

Decoding algorithm with unlikely decoding failure

Proposition

Let $u, v, r \in R$ such that $ug + vs = r$, $hu = \kappa$, $hv = \lambda$ and $hr = \omega$ for some $h \in R$. Let $T = \{l_1, l_2, \ldots, l_m\} = \{0 \leq l \leq n-1 \mid (x - \alpha_l) \mid_r v\}$. Then $m = \deg v$ if and only if $\deg h = 0$.

1. Apply LEEA to compute u_l, v_l, r_l such that

$$u_l g + v_l s = r_l$$

   where $\deg(r_l) < t \leq \deg(r_{l-1})$.
2. Find roots $T = \{l_1, l_2, \ldots, l_m\}$ of v_l in the positional points.
3. if $m = \deg(v_l)$, the error positions are those in T, use r_l to compute error values.

Solving decoding failures

Proposition

Let $u, v, r \in R$ such that $ug + vs = r$, $hu = \kappa$, $hv = \lambda$ and $hr = \omega$ for some $h \in R$. Let $k \in \{0, \dots, n-1\}$ such that $x - \alpha_k \nmid_r v$ but $x - \alpha_k \mid_r \lambda$. Set $v' = [x - \alpha_k, v]_\ell$ and let $h'' \in R$ such that $h''v = v'$. Define $u' = h''u$ and $r' = h''r$. Then $u'g + v's = r'$, $h'u' = \kappa$, $h'v' = \lambda$ and $h'r' = \omega$ for some $h' \in R$.

Proposition

Assume $\lambda = hv$ with $\deg h \geq 1$. Let $\{s_1, \dots, s_m\} = \{i \in \{0, \dots, n-1\} \mid (x - \alpha_i) \mid_r v\}$ and $\{l_1, \dots, l_r\} = \{0, \dots, n-1\} \setminus \{s_1, \dots, s_m\}$. For any $1 \leq i \leq r$, let $f_i = [f_{i-1}, x - \alpha_{l_i}]_\ell$ with $f_0 = v$. Then:

1. *There exists $d \geq 0$ such that $\deg(f_{d-1}) = \deg(f_d)$,*
2. *If d_0 is the minimal index such that $\deg(f_{d_0-1}) = \deg(f_{d_0})$, then $d_0 \in \{k_1, \dots, k_\nu\}$.*

Index

Parameters

Our construction involves skew Goppa polynomials (i.e. $\partial = 0$) over finite fields.

Initial parameters: n, t, q with $t \ll n$, $q = p^d$, $F = \mathbb{F}_q$.

Additional parameters: m, δ, s, μ, where $L = \mathbb{F}_{q^m}$, $\sigma = \tau^s$ (i.e. $\sigma(a) = a^{p^s}$), $\mu = |\sigma|$ and $\delta = \frac{dm}{\mu}$, so that $L^\sigma = K = \mathbb{F}_{p^\delta}$.

Restrictions on the parameters

Proposition

Let $\gamma \in L$ a primitive element. Every maximal P-indepedent subset of L^* if of the form

$$\{\sigma(c_{ij})\gamma^i c_{ij}^{-1} : i = 0, \dots, p^\delta - 2, j = 0, \dots, \mu - 1\},$$

where $\{c_{i0}, \dots, c_{i\mu-1}\}$ is a K-basis of L for each $i = 0, \dots, p^\delta - 1$. As a consequence, if a P-independent subset of L^* has n elements, then $n \leq (p^\delta - 1)\mu$.

Since $\mu = \frac{dm}{\delta}$ and $\delta = \gcd(s, dm)$,

$$\max\left\{\frac{n}{10t}, \frac{n\delta}{d(p^\delta - 1)}\right\} \leq m \leq \frac{n}{4t} \text{ and } \delta \mid dm. \tag{7}$$

Parameters examples

If $n = 4096, t = 25, q = p^d = 2$, we get the following combinations:

m	24	26	28	30	32	33	34	36	36	38	39	40
δ	12	13	14	15	16	11	17	12	18	19	13	20

If $n = 2560, t = 22, q = p^d = 2^4$, we get 65 different combinations, where $12 \leq m \leq 29$ and $12 \leq \delta \leq 58$.

Maximal P-independent sets

- Let $\{\alpha, \sigma(\alpha), \ldots, \sigma^{\mu-1}(\alpha)\}$ be a normal basis of L/K. For $0 \leq i \leq \mu - 1$, set $\beta_i = \sigma^{i+1}(\alpha)\sigma^i(\alpha)^{-1}$. Proposition implies that

$$\left\{\gamma^i \beta_j \mid 0 \leq i \leq p^\delta - 2, 0 \leq j \leq \mu - 1\right\}$$

  is a maximal P-independent set of L^*.

- The probability ρ of picking and element which generates a normal basis is bounded from below by

$$\rho \geq \frac{1}{e \left\lceil \log_{p^\delta} \mu \right\rceil}.$$

- There are randomized algorithms with costs $\mathcal{O}(\mu^2 + \mu \log p^\delta)$ and $\mathcal{O}(\mu^{1.82} \log p^\delta)$ to check in a finite field if an element generates a normal basis. In our experiments we have just used the classical Hensel test, which says that for a given $\alpha \in L = \mathbb{F}_{p^{dm}}$, $\left\{\alpha, \alpha^{p^\delta}, \ldots, \alpha^{p^{(\mu-1)\delta}}\right\}$ is a normal basis if and only if $\gcd\left(z^\mu - 1, \alpha z^{\mu-1} + \alpha^{p^\delta} z^{\mu-2} + \cdots + \alpha^{p^{(\mu-2)\delta}} x + \alpha^{p^{(\mu-1)\delta}}\right) = 1$.

- Primitive elements are handled in a similar way. The number of primitive elements in L is $\varphi(|L| - 1) = \varphi(p^{dm} - 1)$ and $\varphi(p^{dm} - 1)/(p^{dm} - 1)$ is asymptotically bounded from below by a constant multiple of $\log\log(p^{dm} - 1)$.

Key shedule I

Private key

- A maximal set of left P-independent elements is

$$\mathsf{P} = \left\{\sigma^{j+1}(\alpha)\gamma^i\sigma^j(\alpha)^{-1} \mid 0 \leq i \leq p^\delta - 2, 0 \leq j \leq \mu - 1\right\}$$

  for a primitive γ and a normal α randomly computed.
- The list E of positional points is obtained by a random selection of n points in P.

$$\mathsf{E} = \{\alpha_0, \ldots, \alpha_{n-1}\} \subseteq \mathsf{P}.$$

- For the skew Goppa polynomial, we randomly choose a monic polynomial $h(y) \in K[y]$ without roots in K such that $\deg_y(h) = \lfloor 2t/\mu \rfloor$ and set $g = h(x^\mu)x^{2t \bmod \mu}$, which has degree $2t$.
- The REEA allow to compute $h_0, \ldots, h_{n-1} \in R$ such that, for each $0 \leq i \leq n-1$, $\deg(h_i) < 2t$ and

$$(x - \alpha_i)h_i - 1 \in Rg.$$

  In fact $\deg(h_i) = 2t - 1$ by a degree argument.

Key shedule II

Public key

- A parity check matrix for the skew Goppa code is

$$H = \left(\mathfrak{v}(\sigma^{-j}(h_{i,j})\eta_i)\right)_{\substack{0\le j\le 2t-1\\ 0\le i\le n-1}} \in F^{(2tm)\times n}$$

  where $h_i = \sum_{j=0}^{2t-1} h_{i,j}x^j$.
- Once H is computed, the public key of our cryptosystem can be computed as follows: set $k = n - 2t\left\lfloor \frac{n}{4t} \right\rfloor$, $r_H = \mathrm{rank}(H)$ and $A \in F^{(n-k-r_H)\times n}$, a random full rank matrix. The matrix H_{pub} is formed by the non zero rows of the reduced row echelon form of the block matrix $\binom{H}{A}$. If H_{pub} has less that $n-k$ rows, pick a new A. This H_{pub} defines randomly a linear subcode of $\mathcal{C}$ of dimension k.

Key Encapsulation Mechanism Encryption

We pick a random error vector, i.e. $e \in F^n$ such that $\mathrm{w}(e) \leq t$, with corresponding error polynomial $e(x) = \sum_{j=1}^{\nu} e_j x^{k_j}$, $\nu \leq t$ and $0 \leq k_1, k_2, \cdots, k_\nu \leq n-1$. The sender can easily derive a shared secret key from e by means of a fixed and publicly known hash function $\mathcal{H}$. The cryptogram is

$$s = eH_{\mathrm{pub}}^{\mathsf{T}} \in F^{n-k}.$$

Key Encapsulation Mechanism Decryption

The receiver can easily compute $y \in F^n$ such that

$$s = yH_{\text{pub}}^{\mathsf{T}}$$

since H_{pub} is in row reduced echelon form. Let $y(x) = \sum_{i=0}^{n-1} y_i x^i \in R$. Decoding algorithm can be applied to y in order to compute e. Then the shared secret key can be retrieved by the receiver as $\mathcal{H}(e)$.

Remark

Decoding algorithm has unlikely decoding failures. If one of these decoding failures appears, the solving decoding failure subroutine can be applied or a new shared secret can be requested.

Thank you for your attention!

(Gracias por no roncar demasiado fuerte)