Post-Quantum Cryptography

Paulo S. L. M. Barreto

LARC/PCS/EPUSP

Syndrome Decoding

Syndrome Decoding

- Let $q = p^m$ for some prime p and $m > 0$ (for cryptographic applications $p = 2$).
- The (Hamming) *weight* $\mathrm{w}(u)$ of $u \in (\mathbb{F}_q)^n$ is the number of nonzero components of u.
- The distance between $u, v \in (\mathbb{F}_q)^n$ is $\mathrm{dist}(u, v) \equiv \mathrm{w}(u - v)$.
- A linear $[n, k]$-*code* $\mathcal{C}$ over $\mathbb{F}_q$ is a k-dimensional vector subspace of $(\mathbb{F}_q)^n$.

General/Syndrome Decoding (GDP/SDP)

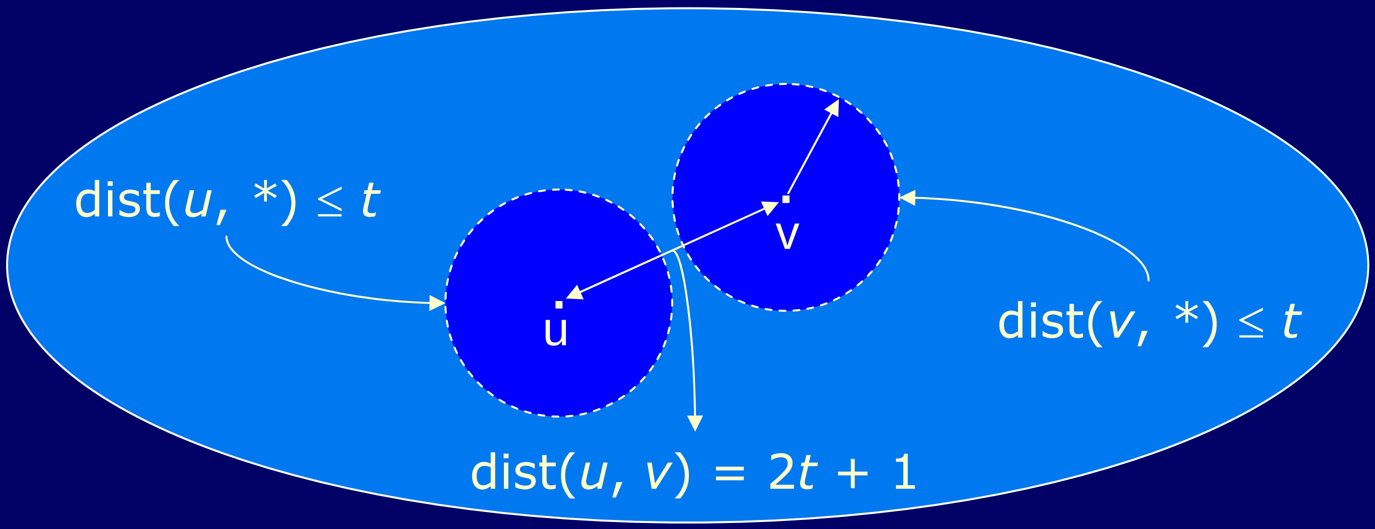
- **GDP**
- **Input:**
  - positive integers n, k, t;
  - generator matrix $G \in (\mathbb{F}_q)^{k \times n}$;
  - vector $c \in (\mathbb{F}_q)^n$.
- **Question:** $\exists$? $m \in (\mathbb{F}_q)^k$ such that $e = c - mG$ has weight $w(e) \le t$?

- **SDP**
- **Input:**
  - positive integers n, r, t;
  - parity-check matrix $H \in (\mathbb{F}_q)^{r \times n}$;
  - vector $s \in (\mathbb{F}_q)^r$.
- **Question:** $\exists$? $e \in (\mathbb{F}_q)^n$ of weight $w(e) \le t$ such that $He^\mathsf{T} = s^\mathsf{T}$?

Both are NP-complete!

Syndrome Decoding

- Let $d = \min\{\mathrm{dist}(u, v) \mid u, v \in \mathcal{C}\}$. If $v, e \in (\mathbb{F}_2)^n$ and $\mathrm{w}(e) \leq \lfloor (d-1)/2 \rfloor \equiv t$, the SDP has a unique solution for $c = v \oplus e$.



Syndrome Decoding

- Determining the minimum distance of a linear code is *NP*-hard.
- Bounded Distance Decoding Problem (BDDP):
  - Given a binary (n, k)-code $\mathcal{C}$ with known minimum distance d and $c \in (\mathbb{F}_2)^n$, find $v \in \mathcal{C}$ such that $\mathrm{dist}(v, c) = d$.
- $\therefore$BDDP is SDP with knowledge of d.
- BDDP is *believed* (but not known for sure) to be intractable.

Ranking and Unranking Permutations

- Some SDP-based cryptosystems represent messages as t-error n-vectors, i.e. n-bit vectors with Hamming weight t.
- Mapping messages between error vector and normal form involves permutation *ranking* and *unranking*.

Ranking and Unranking Permutations

- Let $B(n, t) = \{u \in (\mathbb{F}_2)^n \mid \mathrm{w}(u) = t\}$, with cardinality

$$r = \binom{n}{t} \approx \frac{n^t}{t!}$$

- A *ranking function* is a mapping *rank*: $B(n, t) \to \{1 \ldots r\}$ which associates a unique index in $\{1 \ldots r\}$ to each element in $B(n, t)$. Its inverse is called the *unranking function*.
- Rank size: $\lg r \approx t\,(\lg n - \lg t + 1)$ bits.

Ranking and Unranking Permutations

- Ranking and unranking can be done in $O(n)$ time (Ruskey 2003, algorithm 4.10).
- Computationally simplest ordering: colex.
- Definition: $a_1a_2\ldots a_n < b_1b_2\ldots b_m$ in colex order iff $a_n\ldots a_2a_1 < b_m\ldots b_2b_1$ in lex order.

Colex Ranking

- Sum of binomial coefficients:

$$Rank(a_1 a_2 \ldots a_k) = \sum_{j=1}^{k} \binom{a_j - 1}{j}$$

- Implementation strategy: precompute a table of binomial coefficients.

Colex Unranking

input: r // permutation rank
for $j \leftarrow k$ **downto** 1 {
  $p \leftarrow j$
  while $\binom{p}{j} \le r$ {
    $p \leftarrow p + 1$
  }
  $r \leftarrow r - \binom{p-1}{j}$
  $a_j \leftarrow p$
}
return $a_1\ a_2 \ldots a_k$

Irreducible Polynomials

- Theorem: for $i \geq 1$, the polynomial $x^{q^i} - x \in \mathbb{F}_q[x]$ is the product of all monic irreducible polynomials in $\mathbb{F}_q[x]$ whose degree divides i.
- Ben-Or irreducibility test: monic $g \in \mathbb{F}_q[x]$ of degree d is irreducible iff $\mathrm{GCD}(g, x^{q^i} - x \bmod g) = 1$ for $i = 1, \ldots, d/2$.

Irreducible Polynomials

- Efficient implementation of Ben-Or:
  - compute $y \leftarrow x^q \bmod g$.
  - compute $z_i \leftarrow y^i \bmod g$ for $0 \leq i < t$.
  - initialize $v \leftarrow x$.
  - for $j = 1, \ldots, t/2$:
    - let $v = \sum_{i=0}^{t-1} v_i x^i$: set $v \leftarrow x^{q^j} \bmod g = v^q \bmod g = (\sum_{i=0}^{t-1} v_i x^i)^q \bmod g = \sum_{i=0}^{t-1} v_i (x^q \bmod g)^i \bmod g = \sum_{i=0}^{t-1} v_i (y^i \bmod g) = \sum_{i=0}^{t-1} v_i z_i$.
    - check that $\mathrm{GCD}(g, (v - x) \bmod g) \neq 1$.

Goppa Codes

- Let $g(x) = \sum_{i=0}^{t} g_i x^i$ be a monic ($g_t = 1$) polynomial in $\mathbb{F}_q[x]$.
- Let $L = (L_0, \ldots, L_{n-1}) \in (\mathbb{F}_q)^n$ (all distinct) such that $g(L_j) \neq 0$ for all j.
- Properties:
  - Easy to generate and plentiful.
  - Usually $g(x)$ is chosen to be irreducible; if so, $\mathbb{F}_{q^t} = \mathbb{F}[x]/g(x)$.

Goppa Codes

- The *syndrome function* is the linear map $S: (\mathbb{F}_p)^n \to \mathbb{F}_q[x]/g(x)$:

$$S(c) = \sum_{i=0}^{n-1} \frac{c_i}{x - L_i} = \sum_{c_i=1} \frac{1}{x - L_i} \pmod{g(x)}.$$

- The *Goppa code* $\Gamma(L, g)$ is the kernel of the syndrome function, i.e. $\Gamma = \{c \in (\mathbb{F}_p)^n \mid S(c) = 0\}$.

Goppa Codes

- N.B. Usually $t = O(n / \lg n)$. CFS are an exception, with $n = O(t!)$.
- The syndrome can be written in matrix form as a mapping $H^*: (\mathbb{F}_p)^n \to (\mathbb{F}_q)^t$ or even $H: (\mathbb{F}_p)^n \to (\mathbb{F}_p)^{mt}$ (just write the $\mathbb{F}_p$ components of each $\mathbb{F}_q$ element from H^* on m successive rows of H).
- H is the *parity check matrix* of the code. Determining whether $c \in (\mathbb{F}_p)^n$ is a code word amounts to checking that $Hc^T = 0$.

Parity-Check Matrix

- Easy to compute H^* from L and g, namely, $H^*_{t\times n} = T_{t\times t}V_{t\times n}D_{n\times n}$, where:

$$T = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ g_{t-1} & 1 & 0 & \dots & 0 \\ g_{t-2} & g_{t-1} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_1 & g_2 & g_3 & \dots & 1 \end{bmatrix}, \quad V = \begin{bmatrix} 1 & 1 & \dots & 1 \\ L_0 & L_1 & \dots & L_{n-1} \\ L_0^2 & L_1^2 & \dots & L_{n-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ L_0^{t-1} & L_1^{t-1} & \dots & L_{n-1}^{t-1} \end{bmatrix},$$

$$D = \begin{bmatrix} 1/g(L_0) & 0 & \dots & 0 \\ 0 & 1/g(L_1) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1/g(L_{n-1}) \end{bmatrix}.$$

Generator Matrix

- A Goppa code Γ is a k-dimensional subspace of $(\mathbb{F}_p)^n$ for some k with $n - mt \le k \le n - t$.
- In general the minimum distance of Γ is $d \ge t + 1$, but in the *binary* case whenever $g(x)$ has no multiple zero (in particular when $g(x)$ is irreducible) the minimum distance becomes $d \ge 2t + 1$.

Generator Matrix

- A *generator matrix* for Γ is a matrix $G_{k \times n}$ whose rows form a basis of Γ.
- G defines a mapping $(\mathbb{F}_p)^k \to (\mathbb{F}_p)^n$ such that $uG \in \Gamma$, $\forall u \in (\mathbb{F}_p)^k$.
- Therefore $H(uG)^T = HG^T u^T = o^T$ for all u, i.e. $HG^T = O$.

Generator Matrix

- If G is in *echelon* form, it is trivial to map between $(\mathbb{F}_p)^k$ and $(\mathbb{F}_p)^n$.
- The first k columns of $uG \in (\mathbb{F}_p)^n$ directly spell $u \in (\mathbb{F}_p)^k$ itself.
- The remaining $n - k$ columns contain the "checksum" of u.

Generator Matrix

- It is easy to solve $H_{mt\times n}G^{\mathrm{T}}_{n\times k} = O_{mt\times k}$ for G in echelon form and $k = n - mt$, i.e. $G_{k\times n} = [I_{k\times k} \mid X_{k\times mt}]$.
- Let $H_{mt\times n} = [L_{mt\times k} \mid R_{mt\times mt}]$. Equation $HG^{\mathrm{T}} = O$ becomes $[L_{mt\times k} \mid R_{mt\times mt}]\ [I_{k\times k} \mid X^{\mathrm{T}}_{mt\times k}] = L_{mt\times k} + R_{mt\times mt}X^{\mathrm{T}}_{mt\times k} = O_{mt\times k}$, whose solution is $X^{\mathrm{T}}_{mt\times k} = R^{-1}_{mt\times mt}L_{mt\times k}$, or $G_{k\times n} = [I_{k\times k} \mid L^{\mathrm{T}}_{k\times mt}\,(R^{\mathrm{T}})^{-1}_{mt\times mt}]$.

Generator Matrix

- Any nonzero matrix H' satisfying $H'G^{\mathrm{T}} = O$ is an alternative parity check matrix.
  - Since $T_{t\times t}$ is invertible ($\det(T) = 1$) and $H_{t\times n} = T_{t\times t}V_{t\times n}D_{n\times n}$, clearly $H'G^{\mathrm{T}} = O$ for $H' = VD$.
  - Let $G_{k\times n} = [I_{k\times k} \mid X_{k\times t}]$ and $H'' = [X^{\mathrm{T}}_{t\times k} \mid I_{t\times t}]$. Clearly $[X^{\mathrm{T}}_{t\times k} \mid I_{t\times t}]\ [I_{k\times k} \mid X^{\mathrm{T}}_{t\times k}] = O_{t\times k}$, i.e. $H''G^{\mathrm{T}} = O$.
  - For any nonsingular matrix $S_{t\times t}$, $H''' \leftarrow SH''$ satisfies $H'''G^{\mathrm{T}} = O$.

Error Correction

Error Locator Polynomial

- Efficient decoding procedure for known g and L via the *error locator polynomial*:

$$\sigma(x) \equiv \prod_{e_i=1} (x - L_i) \in \mathbb{F}_q[x]/g(x).$$

- Property: $\sigma(L_i) = 0 \Leftrightarrow e_i = 1$.

Alternant Error Locator Polynomial

- Efficient decoding procedure for known g and L via the *error locator polynomial*:

$$\sigma(x) \equiv \prod_{e_i \neq 0} (1 - xL_i) \in \mathbb{F}_q[x]/g(x).$$

- Property: $\sigma(L_i^{-1}) = 0 \Leftrightarrow e_i \neq 0.$

Error Correction

- Let $m \in \Gamma$, let $e \in (\mathbb{F}_2)^n$ be an error vector of weight $w(e) \le t$, and $c = m \oplus e$.
- Compute the syndrome of e through the relation $S(e) = S(c)$.
- Compute the error locator polynomial σ from the syndrome (Sugiyama *et al*. 1975).
- Determine which L_i are zeroes of σ, thus retrieving e and recovering m.

Error Correction (aka "Binary Goppa Miracle")

- Let $s(x) \leftarrow S(e)$. If $s(x) \equiv 0$, nothing to do (no error), otherwise $s(x)$ is invertible.
  - Property #1: $\sigma(x) = a(x)^2 + xb(x)^2.$
  - Property #2: $\frac{d}{dx}\sigma(x) = b(x)^2.$
  - Property #3: $\frac{d}{dx}\sigma(x) = \sigma(x)s(x).$
- Thus $b(x)^2 = (a(x)^2 + xb(x)^2)s(x)$, hence $a(x) = b(x)v(x)$ with $v(x) = \sqrt{x + 1/s(x)} \bmod g(x)$.

$\underbrace{a(x) = b(x)v(x)}_{\text{Extended Euclid!}}$ … $\underbrace{1/s(x)}_{\text{Extended Euclid!}}$

Extended Euclid! Extended Euclid!

Computing $s(x)^{-1} \pmod{g(x)}$

$F \leftarrow s,\ G \leftarrow g,\ B \leftarrow 1,\ C \leftarrow 0$

while (deg(F) > 0) {

    if (deg(F) < deg(G)) {

        $F \leftrightarrow G,\ B \leftrightarrow C$

    }

    $j \leftarrow \deg(F) - \deg(G),\ h \leftarrow F_{\deg(F)} / G_{\deg(G)}$

    $F \leftarrow F - h\, x^j\, G,\ B \leftarrow B - h\, x^j\, C$

}

if ($F \neq 0$) **return** B / F_0 **else** "not invertible"

Decoding a binary Goppa syndrome $s(x)$

- Given: $v(x)$, $g(x) \in \mathbb{K}[x]$
- Find: $a(x)$, $b(x)$, $f(x) \in \mathbb{K}[x]$
- Where: $b(x)v(x) + f(x)g(x) = a(x)$
- Thus $a(x) = b(x)v(x) \bmod g(x)$, i.e. $a(x) = b(x)v(x)$ in $\mathbb{K}[x]/g(x)$.
- Conditions:
  - $\deg(a) \le \lfloor t/2 \rfloor$, $\deg(b) \le \lfloor (t-1)/2 \rfloor$.

Decoding a binary Goppa syndrome $s(x)$

$A \leftarrow v,\ a \leftarrow g,\ B \leftarrow 1,\ b \leftarrow 0,\ t \leftarrow \deg(g)$

while $(\deg(a) > \lfloor t/2 \rfloor)$ {

$\quad A \leftrightarrow a,\ B \leftrightarrow b$

$\quad$ **while** $(\deg(A) \geq \deg(a))$ {

$\quad\quad j \leftarrow \deg(A) - \deg(a),\ h \leftarrow A_{\deg(A)} / a_{\deg(a)}$

$\quad\quad A \leftarrow A - h\, x^j a,\ B \leftarrow B - h\, x^j b$

$\quad$ }

}

$\sigma(x) \leftarrow a(x)^2 + x b(x)^2$

return σ // error locator polynomial

Decoding an alternant syndrome $s(x)$

- Given: $s(x) \in \mathbb{K}[x]$, $t \in \mathbb{N}$
- Find: $\omega(x)$, $\sigma(x)$, $f(x) \in \mathbb{K}[x]$
- Where: $\sigma(x)s(x) + f(x)x^{2t} = \omega(x)$
- Thus $\omega(x) = \sigma(x)s(x) \bmod x^{2t}$, i.e. $\omega(x) = \sigma(x)s(x) \in \mathbb{K}[x]/x^{2t}$.
- Conditions:
  - $\deg(\omega) \le t - 1$, $\deg(\sigma) \le t$.

Decoding an alternant syndrome $s(x)$

$A \leftarrow s,\ a \leftarrow x^{2t},\ B \leftarrow 1,\ b \leftarrow 0$
while (deg(a) > t – 1) {
  $A \leftrightarrow a,\ B \leftrightarrow b$
  while (deg(A) ≥ deg(a)) {
    $j \leftarrow \deg(A) - \deg(a),\ h \leftarrow A_{\deg(A)} / a_{\deg(a)}$
    $A \leftarrow A - h\, x^j a,\ B \leftarrow B - h\, x^j b$
  }
}
$\sigma(x) \leftarrow b(x) / b_0$ // hence $\sigma(0) = 1$
$\omega(x) \leftarrow a(x) / b_0$ // normalize
return ω, σ // error evaluator & locator polynomials

Coding-Based Cryptosystems

McEliece Cryptosystem

- Key generation:
  - Let p be a prime power and $q = p^d$ for some d.
  - Choose a secure, uniformly random $[n, k]$ t-error correcting alternant code $\mathcal{A}(L, D)$ over $\mathbb{F}_p$, with $L, D \in (\mathbb{F}_q)^n$.
  - N.B. $\mathcal{A}(L, D)$ defined e.g. by the parity-check matrix $H = \mathrm{vdm}(L)\ \mathrm{diag}(D)$.
  - Compute for $\mathcal{A}(L, D)$ a systematic generator matrix $G \in (\mathbb{F}_p)^{k \times n}$.
  - Set $K_{priv} = (L, D)$, $K_{pub} = (G, t)$.

McEliece Cryptosystem

- "Hey, wait, I know McEliece, and this does not look quite like it!"
- Observations:
  - A *secret*, *random* L is equivalent to a *public*, *fixed* L coupled to a *secret*, *random* permutation matrix $P \in (\mathbb{F}_p)^{k \times k}$, with $\mathcal{A}(LP, DP)$ as the effective code.
  - If G_0 is a generator for $\mathcal{A}(L, D)$ when L is public and fixed, and S is the matrix that puts G_0P in systematic form, then $G = SG_0P$ is a systematic generator of $\mathcal{A}(LP, DP)$, as desired.
  - Goppa: $D = 1/g(L)$, $\mathcal{A}(L, D) = \Gamma(L, g)$, $\mathrm{K}_{\mathrm{priv}} = (L, g)$.

McEliece Cryptosystem

- Encryption of a plaintext $m \in (\mathbb{F}_p)^k$:
  - Choose a uniformly random t-error vector $e \in (\mathbb{F}_p)^n$ and compute $c = mG + e \in (\mathbb{F}_p)^n$ (IND-CCA2 variant via e.g. Fujisaki-Okamoto).
- Decryption of a ciphertext $c \in (\mathbb{F}_p)^n$:
  - Use the trapdoor to obtain the usual alternant parity-check matrix H (or equivalent).
  - Compute the syndrome $s^T \leftarrow Hc^T = He^T$ and decode it to obtain the error vector e.
  - Read m directly from the first k components of $c - e$.

McEliece-Fujisaki-Okamoto: Setup

- Random oracle (message authentication code) $\mathcal{H}$: $(\mathbb{F}_p)^k \times \{0, 1\}^* \to \mathbb{Z}/s\mathbb{Z}$, with s = (n choose t) $(p - 1)^t$.
- Unranking function $\mathcal{U}$: $\mathbb{Z}/s\mathbb{Z} \to (\mathbb{F}_p)^n$.
- Ideal symmetric cipher $\mathcal{E}$: $(\mathbb{F}_p)^k \times \{0, 1\}^* \to \{0, 1\}^*$.
- Alternant decoding algorithm $\mathcal{D}$: $(\mathbb{F}_q)^n \times (\mathbb{F}_q)^n \times (\mathbb{F}_p)^n \to (\mathbb{F}_p)^k \times (\mathbb{F}_p)^n$.

McEliece-Fujisaki-Okamoto: Encryption

- Input:
  - uniformly random symmetric key $r \in (\mathbb{F}_p)^k$;
  - message $m \in \{0, 1\}^*$.
- Output:
  - McEliece-FO ciphertext $c \in (\mathbb{F}_p)^n \times \{0, 1\}^*$.
- Algorithm:
  - $h \leftarrow \mathcal{H}(r, m)$
  - $e \leftarrow \mathcal{U}(h)$
  - $w \leftarrow rG + e$
  - $d \leftarrow \mathcal{E}(r, m)$
  - $c \leftarrow (w, d)$

McEliece-Fujisaki-Okamoto: Decryption

- Input:
  - McEliece-FO ciphertext $c = (w, d)$.
- Output:
  - message $m \in \{0, 1\}^*$, or rejection.
- Algorithm:
  - $(r, e) \leftarrow \mathcal{D}(L, D, w)$
  - $m \leftarrow \mathcal{E}^{-1}(r, d)$
  - $h \leftarrow \mathcal{H}(r, m)$
  - $v \leftarrow \mathcal{U}(h)$
  - accept $m \Leftrightarrow v = e$ and $w = rG + e$

Niederreiter Cryptosystem

- Key generation:
  - Choose a secure, uniformly random $[n, k]$ t-error correcting alternant code $\mathcal{A}(L, D)$ over $\mathbb{F}_p$, with $L, D \in (\mathbb{F}_q)^n$.
  - Compute for $\mathcal{A}(L, D)$ a systematic parity-check matrix $H \in (\mathbb{F}_p)^{r \times n}$.
  - Set $K_{\text{priv}} = (L, D)$, $K_{\text{pub}} = (H, t)$.

Niederreiter Cryptosystem

- Encryption of plaintext $m \in \mathbb{Z}/s\mathbb{Z}$, $s = (n \text{ choose } t)\,(p - 1)^t$:
  - Represent m as a t-error vector $e \in (\mathbb{F}_p)^n$ via permutation unranking.
  - Compute the syndrome $c^\mathsf{T} = He^\mathsf{T}$ as ciphertext.
- Decryption of ciphertext $c \in (\mathbb{F}_p)^r$:
  - Let $H_0 = \mathrm{vdm}(L)\,\mathrm{diag}(D)$ be the trapdoor parity-check matrix for $\mathcal{A}(L, D)$, so that $H_0 = SH$ for some nonsingular matrix S. Compute $c_0^\mathsf{T} = Sc^\mathsf{T}$. Notice that $c_0^\mathsf{T} = S(He^\mathsf{T}) = H_0e^\mathsf{T}$, a decodable syndrome (using the trapdoor). Also, $S = H_0H^\mathsf{T}(HH^\mathsf{T})^{-1}$.
  - Decode the syndrome c_0^T to e^T using the decoding trapdoor.
  - Recover m from e via permutation ranking.

Niederreiter Cryptosystem

- The computational security levels of McEliece and Niederreiter are exactly equivalent.
- Both need extra message formatting to achieve indistinguishability properties.
- Niederreiter leads more naturally to digital signatures.

CFS Signatures

- Security based on the BDDP assumption.
- Represent the message as a decodable syndrome, then decode the syndrome to produce the error vector as the signature.
- Verify the signature by matching it to the syndrome of the message.
- Short signatures possible via permutation ranking.

CFS Signatures

- System setup:
  - Choose m, $t \leq m$ and $n = 2^m$.
  - Choose a hash function $\mathcal{H}$: $\{0, 1\}^* \times \mathbb{N} \to (\mathbb{F}_2)^{n-k}$.
- Key generation:
  - Choose a t-error correcting, binary Goppa code $\Gamma(L, g)$, compute for it a systematic parity-check matrix H.
  - $K_{private} = (L, g)$; $K_{public} = (H, t)$.

CFS Signatures

- Signing a message m:
  - Let H_0 be the trapdoor parity-check matrix for $\Gamma(L, g)$, so that $H_0 = SH$ for some nonsingular matrix S. Find $i \in \mathbb{N}$ such that, for $c \leftarrow \mathcal{H}(m, i)$ and $c_0^\mathsf{T} \leftarrow Sc^\mathsf{T}$, c_0 is a decodable H_0-syndrome of Γ.
  - Using the decoding algorithm for Γ, compute the error vector e whose H_0-syndrome is c_0, i.e. $c_0^\mathsf{T} = H_0 e^\mathsf{T}$.
  - The signature is (e, i). Notice that $c_0^\mathsf{T} = H_0 e^\mathsf{T} = SHe^\mathsf{T}$ and hence $He^\mathsf{T} = S^{-1} c_0^\mathsf{T} = c^\mathsf{T}$, i.e. $c = \mathcal{H}(m, i)$ is the H-syndrome of e.
- Verifying a signature (e, i):
  - Compute $c \leftarrow He^\mathsf{T}$.
  - Accept the signature iff $c = \mathcal{H}(m, i)$.

CFS Signatures

- The number of possible hash values is $2^{n-k} = 2^{mt} = n^t$ and the number of syndromes decodable to codewords of weight t is

$$\binom{n}{t} \approx \frac{n^t}{t!}$$

- $\therefore$The probability of finding a codeword of weight t is $\approx 1/t!$, and the expected value of hash queries is $\approx t!$.

CFS Signatures

- If the n-bit error e of weight t is encoded via permutation ranking, the signature length is $\approx \lg(n^t/t!) + \lg(t!) = t \lg n \approx mt$.
- Public key is huge: mtn bits.
- Recommendation for security level $\approx 2^{80}$:
  - original: $m = 16$, $t = 9$, $n = 2^{16}$, signature length = 144 bits, key size = 1152 KiB.
  - updated: $m = 15$, $t = 12$, $n = 2^{15}$, signature length = 180 bits, key size = 720 KiB;

CFS Signatures

- Bleichenbacher's attack: Wagner's generalized (3-way) birthday attack ⇒ security level lower than expected.
- Larger key sizes, longer signature generation.
- Dyadic keys: shorter by a factor *u* = largest power of 2 dividing *t*, but 2^u times longer signature generation.

m	t=9	t=10	t=11	t=12
15	60.2	63.1	67.2	**81.5**
16	**63.3**	66.2	71.3	**85.6**
17	66.4	69.3	75.4	**89.7**
18	69.5	72.4	79.5	**93.7**
		...		
22	**81.7**	**84.6**	**95.8**	**110.0**

Stern Identification

- $H \in (\mathbb{F}_2)^{(n/2)\times n}$: uniformly random binary parity-check matrix (N.B. originally of size $(n-k) \times n$).
- Gaborit-Girault improvement: uniformly random double circulant $H = [I \mid C]$, with $C_{ij} = c_{(j-i) \bmod n/2}$ for some $c \in (\mathbb{F}_2)^{n/2}$.
- Misoczki-Barreto alternative: uniformly random double dyadic $H = [I \mid D]$, with $D_{ij} = d_{i \oplus j}$ for some $d \in (\mathbb{F}_2)^{n/2}$.

Stern Identification

- Key pair:
  - Private key: random $x \in (\mathbb{F}_2)^n$ of weight t.
  - Public key: syndrome $s = xH^T \in (\mathbb{F}_2)^{n/2}$.

Stern Identification

- Commitment:
  - The prover chooses a uniformly random word $y \in (\mathbb{F}_2)^n$ and a uniformly random permutation σ on $\{0, \ldots, n-1\}$ and sends $c_0 = \text{hash}(\sigma(y))$, $c_1 = \text{hash}(\sigma(y + x))$, and $c_2 = \text{hash}(\sigma \mid\mid Hy^\mathsf{T})$ to the verifier.

Stern Identification

- Challenge & Response:
  - The verifier sends a uniformly random $b \in \mathbb{F}_3$ to the prover.
  - The prover responds by revealing:
    - y and σ if $b = 0$;
    - $y + x$ and σ if $b = 1$;
    - $\sigma(y)$ and $\sigma(x)$ if $b = 2$.

Stern Identification

- Verification:
  - The verifier verifies that:
    - c_0 and c_2 are correct if $b = 0$;
    - c_1 and c_2 are correct if $b = 1$ (noticing that $Hy^T = H(y + x)^T + Hx^T = H(y + x)^T + s^T$);
    - c_0 and c_1 are correct if $b = 3$ (noticing that $\sigma(y + x) = \sigma(y) + \sigma(x)$).
  - The probability of cheating in this ZKP is 2/3. Repeating $\lceil(\lg \varepsilon)/(1 - \lg 3)\rceil$ times reduces the cheating probabiliy below ε.

Stern Identification

- Gaborit-Girault propose $n = 347$ and $t = 76$ to achieve security 2^{83} with double circulant keys.
- Exactly the same parameters are fine with double dyadic keys.
- In either case the key is only $2n = 694$ bits long and the global matrix H fits $n = 347$ bits.

Stern Identification

- Identity-based identification: Goppa trapdoor for the Stern scheme combined with CFS signatures.
- Stern public key is the user's identity mapped to a decodable syndrome (N.B. the identity has to be complemented by a short counter provided by the KGC).
- Identity-based private key is a CFS signature of the user's identity, i.e. an error vector x of weight t computed by the KGC.

Choosing Parameters

- Using systematic (echelon) form, storage reduces to only $k \times (n - k)$ bits.

security level	m	n	k	t	naïve key size	echelon key size	source
2^{56}	10	1024	524	50	65.5 KiB	32 KiB	original
2^{80}	11	1632	1269	33+1	74–253 KiB	57 KiB	BLP
2^{112}	12	2480	1940	45+1	164–587 KiB	128 KiB	BLP
2^{128}	12	2960	2288	56+1	243–827 KiB	188 KiB	BLP
2^{192}	13	4624	3389	95+2	698–1913 KiB	511 KiB	BLP
2^{256}	13	6624	5129	115+2	1209–4147 KiB	937 KiB	BLP

Choosing the Code

- Most syndrome-based cryptosystems can be instantiated with general (n, k)-codes.
- Not all choices of code are secure.
  - McEliece with maximum rank distance (MRD) or Gabidulin codes is insecure (Gibson 1995, 1996).
  - Niederreiter with GRS codes is insecure (Sidelnikov-Shestakov 1992).
- Binary Goppa seems to be OK.
  - … Except if the coefficients of the Goppa polynomial itself are all binary (Loidreau-Sendrier 1998).
  - Distinguishing a (complete) permuted Goppa code from a random code of the same length and distance (Sendrier 2000): $O(t\, n^{t-2} \log^2 n)$.

Compact Goppa Codes?

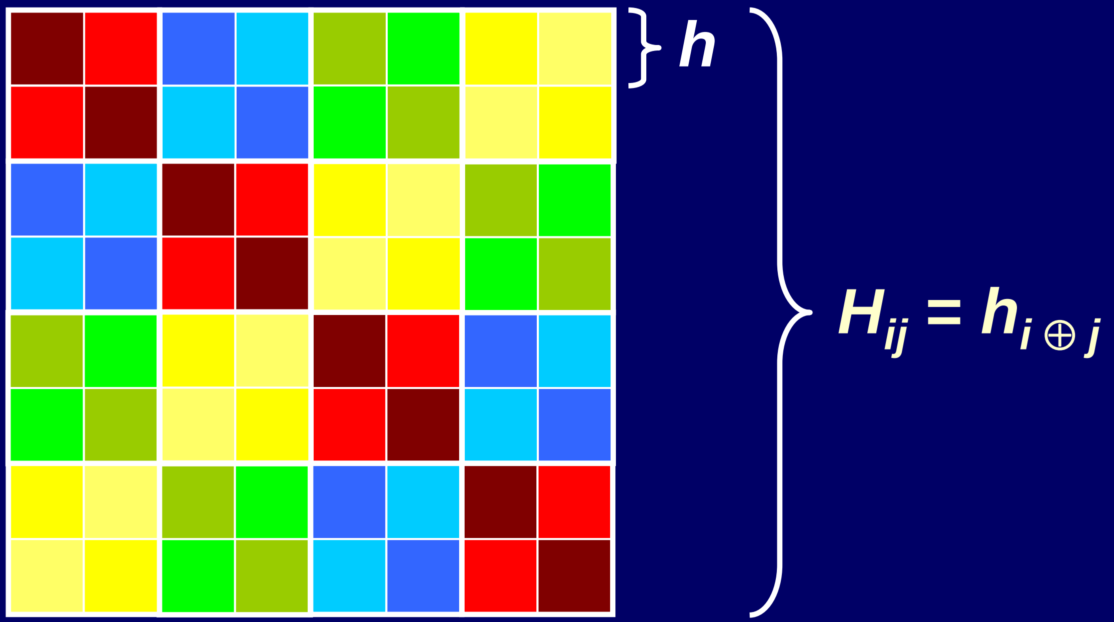
- Recap: a *Goppa code* is entirely defined by:
  - a monic polynomial $g(x) \in \mathbb{F}_q[x]$ of degree t,
  - a sequence $L \in (\mathbb{F}_q)^n$ of distinct elements with $g(L) \neq 0$.
- Features:
  - good error correction capability (all t design errors in the binary case).
  - withstood cryptanalysis quite well.
- Goal: replace the large $O(n^2)$-bit representation by a compact one (like above!).

Cauchy Matrices

- A matrix $M \in \mathbb{K}^{t \times n}$ over a field $\mathbb{K}$ is called a *Cauchy matrix* iff $M_{ij} = 1/(z_i - L_j)$ for disjoint sequences $z \in \mathbb{K}^t$ and $L \in \mathbb{K}^n$ of distinct elements.
- Property: any Goppa code where $g(x)$ is square-free admits a parity-check matrix in Cauchy form [TZ 1975].
- Compact representation, but:
  - code structure is apparent,
  - usual tricks to hide it destroy the Cauchy structure.

Dyadic Matrices

- Let r be a power of 2. A matrix $H \in \mathcal{R}^{r \times r}$ over a ring $\mathcal{R}$ is called *dyadic* iff $H_{ij} = h_{i \oplus j}$ for some vector $h \in \mathcal{R}^r$.



Dyadic Matrices

- Dyadic matrices form a subring of $\mathcal{R}^{r\times r}$ (commutative if $\mathcal{R}$ is commutative).
- Compact representation: O(r) rather than O(r^2) space.
- Efficient arithmetic: multiplication in time O(r lg r) time via fast Walsh-Hadamard transform, inversion in time O(r) in characteristic 2.
- **Idea:** find a dyadic Cauchy matrix.

Quasi-Dyadic Codes

- **Theorem:** a dyadic Cauchy matrix is only possible over fields of characteristic 2 (i.e. $q = 2^m$ for some m), and any suitable $h \in (\mathbb{F}_q)^n$ satisfies

$$\frac{1}{h_{i \oplus j}} = \frac{1}{h_i} + \frac{1}{h_j} + \frac{1}{h_0}$$

with $z_i = 1/h_i + \omega$, $L_j = 1/h_j - 1/h_0 + \omega$ for arbitrary ω, and $H_{ij} = h_{i \oplus j} = 1/(z_i - L_j)$.

Dyadic Cauchy Matrices

- Dyadic: $M_{ij} = h_{i \oplus j}$ for $h \in (\mathbb{F}_q)^n$.
- Cauchy: $M_{ij} = 1/(x_i - y_j)$ for $x, y \in (\mathbb{F}_q)^n$.
- Dyadic matrices are symmetric:

  $1/(x_i - y_j) = 1/(x_j - y_i) \Leftrightarrow y_j = x_i + y_i - x_j \Leftrightarrow$ $-y_j = \alpha + x_j$ (taking $i = 0$ in particular) for some constant $\alpha \Leftrightarrow M_{ij} = 1/(x_i + x_j + \alpha)$ for $x \in (\mathbb{F}_q)^n$.
- Dyadic matrices have constant diagonal:
  - $M_{ii} = 1/(2x_i + \alpha) = h_0 \Leftrightarrow$ all x_i equal (impossible) or char 2.

Dyadic Cauchy Matrices

- Condition $h_{i \oplus j} = 1/(x_i + x_j + \alpha)$ shows that $\alpha = 1/h_0$ (taking $i = j$ in particular), hence $1/h_{i \oplus j} + 1/h_0 = x_i + x_j$, or simply

  $$x_i = 1/h_i + 1/h_0 + x_0$$

  (taking $j = 0$ in particular).
- Thus $1/h_{i \oplus j} + 1/h_0 = x_i + x_j = 1/h_i + 1/h_j$, so necessarily the sequence h satisfies

$$\frac{1}{h_{i \oplus j}} = \frac{1}{h_i} + \frac{1}{h_j} + \frac{1}{h_0}$$

Constructing Dyadic Codes

- Choose distinct h_0 and h_i with $i = 2^u$ for $0 \leq u < \lceil \lg n \rceil$ uniformly at random from $\mathbb{F}_q$, then set

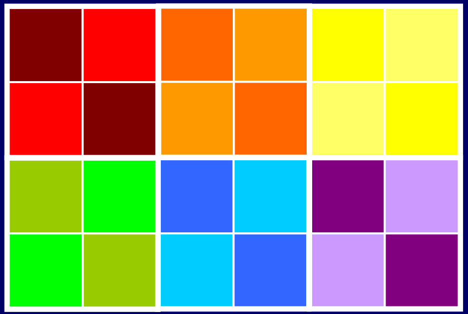
$$h_{i+j} \leftarrow \frac{1}{\frac{1}{h_i} + \frac{1}{h_j} + \frac{1}{h_0}}$$

  for $0 < j < i$ (so that $i + j = i \oplus j$).

- Complexity: O(n).

Quasi-Dyadic Codes

- Structure hiding:
  - choose a long dyadic code over $\mathbb{F}_q$,
  - blockwise shorten the code (Wieschebrink),
  - permute dyadic block columns,
  - dyadic-permute individual blocks,
  - take a binary subfield subcode.
- Quasi-dyadic matrices: $((\mathbb{F}_2)^{t\times t})^{m\times \ell}$.



Compact Keys

- Binary quasi-dyadic codes obtained from a Goppa code over $\mathbb{F}_{2^{16}}$ with $t \times t$ dyadic submatrices:

level	*n*	*k*	*t*	size	generic	shrink	RSA
2^{80}	2304	1280	64	20480 bits	57 KiB	23	1024 bits
2^{112}	3584	1536	128	24576 bits	128 KiB	43	2048 bits
2^{128}	4096	2048	128	32768 bits	188 KiB	47	3072 bits
2^{192}	7168	3072	256	49152 bits	511 KiB	85	7680 bits
2^{256}	8192	4096	256	65536 bits	937 KiB	117	15360 bits

Linear Attacks

- The relation between the decodable private parity-check matrix H and the public generator matrix G is $HXG^T = O$ for some permutation matrix X.
- Attack idea: guess H and solve the above equation for X.
- Possible when (1) it is feasible to guess H, and (2) the linear system is determined.

Linear Attacks

- For a generic, irreducible Goppa code there are roughly $O(q^t/(t \log q)) \sim O(2^{mt}/mt) \sim O(2^{2^m})$ possibilities for H, too many to mount an attack. Besides, X is as general as it can be, so there is no hope of getting a determined linear system.
- For a quasi-cyclic code there are only $O(2^m)$ possibilities. Besides, the linear system is overdetermined due to severe constraints on X. As a consequence, most if not all quasi-cyclic proposals have been broken.

Linear Attacks

- For a quasi-dyadic codes there are $O(2^{m^2})$ possibilities, still too many. Besides, X is only constrained to consist of dyadic submatrices, but these are otherwise independent and the system remains highly indetermined.
- Hence quasi-dyadic, binary Goppa codes resist this kind of attack.