A Simple Introduction to Syndrome-Decoding-Based Cryptography

Paulo S. L. M. Barreto

Contents

- Motivation and basic concepts of error-correcting codes
- Cryptosystems based on syndrome decoding (McEliece and Niederreiter encryption, CFS signatures)
- Constructing and decoding Goppa codes
- Current challenges (reducing key sizes, safe codes, new functionality)

Motivation

Deployed Cryptosystems

- Conventional intractability assumptions:
  - Integer Factorization (IFP): RSA.
  - Discrete Logarithm (DLP), Diffie-Hellman (DHP), bilinear variants: ECC, PBC.

- These assumptions reduce to the *Hidden Subgroup Problem* – HSP.

Quantum Computing

- Shor's quantum algorithm can solve particular cases of the AHSP (including IFP and DLP) in random polynomial time.

⇒ (

=

) ?

Proposed Post-Quantum Cryptosystems

- Quantum computers seem to be unable to solve NP-complete/NP-hard problems.
- Syndrome Decoding (this seminar)
- Lattice Reduction
- Merkle signatures, Multivariate Quadratic Systems, Non-Abelian (e.g. Braid) Groups, Permuted Kernels and Perceptrons, Constrained Linear Equations...

Basic Concepts of Error-Correcting Codes

Linear Codes

- The (Hamming) *weight* $w(u)$ of $u \in (\mathbb{F}_q)^n$ is the number of nonzero components of u, and the (Hamming) distance between u, $v \in (\mathbb{F}_q)^n$ is $\mathrm{dist}(u, v) \equiv w(u - v)$.

- A linear $[n, k]$-*code* $\mathcal{C}$ over $\mathbb{F}_q$ is a k-dimensional vector subspace of $(\mathbb{F}_q)^n$.

Linear Codes

- A code may be defined by a *generator* matrix $G \in (\mathbb{F}_q)^{k\times n}$ or by a *parity-check* matrix $H \in (\mathbb{F}_q)^{r\times n}$ with $r = n - k$.
  - $\mathcal{C} = \{uG \in (\mathbb{F}_q)^n \mid u \in (\mathbb{F}_q)^k\}$
  - $\mathcal{C} = \{v \in (\mathbb{F}_q)^n \mid Hv^T = 0^r\}$

- N.B. The vector s such that $Hv^T = s^T$ is called the *syndrome* of v.
- N.B. $HG^T = O$.

Linear Codes

- Generator and parity-check matrices are not unique: given an arbitrary nonsingular matrix $S \in (\mathbb{F}_q)^{k \times k}$ (resp. $S \in (\mathbb{F}_q)^{r \times r}$), the matrix $G' = SG$ (resp. $H' = SH$) defines the same code as G (resp. H) in another basis.
- Consequence: systematic (echelon) form $G = [I_k \mid M]$, $H = [-M^T \mid I_r]$ where $M \in (\mathbb{F}_q)^{k \times r}$. N.B.: not always possible.

Linear Codes

- Two codes are (permutation) *equivalent* if they differ essentially by a permutation on the coordinates of their elements.
- Formally, a code $\mathcal{C}'$ generated by G' is equivalent to a code $\mathcal{C}$ generated by G iff $G' = SGP$ for some permutation matrix $P \in (\mathbb{F}_q)^{n \times n}$ and some nonsingular matrix $S \in (\mathbb{F}_q)^{k \times k}$. Notation: $\mathcal{C}' = \mathcal{C}P$.

General Decoding

- **Input:** positive integers n, k, t; a finite field $\mathbb{F}_q$; a linear $[n, k]$-code $\mathcal{C} \in (\mathbb{F}_q)^n$ defined by a generator matrix $G \in (\mathbb{F}_q)^{k \times n}$; a vector $c \in (\mathbb{F}_q)^n$.
- **Question:** is there a vector $m \in (\mathbb{F}_q)^k$ s.t. $e = c - mG$ has weight $w(e) \leq t$?
- NP-complete!
- **Search:** find such a vector e.

Syndrome Decoding

- **Input:** positive integers n, k, t; a finite field $\mathbb{F}_q$; a linear $[n, k]$-code $\mathcal{C} \in (\mathbb{F}_q)^n$ defined by a parity-check matrix $H \in (\mathbb{F}_q)^{r \times n}$ with $r = n - k$; a vector $s \in (\mathbb{F}_q)^r$.
- **Question:** is there a vector $e \in (\mathbb{F}_q)^n$ of weight $w(e) \leq t$ s.t. $He^{\mathsf{T}} = s^{\mathsf{T}}$?
- NP-complete!
- **Search:** find such a vector e.

Easily Decodable Codes

- Some codes allow for efficient decoding, e.g. GRS/alternant codes with a parity-check matrix of form $H = VD$ with

$$V = \begin{bmatrix} 1 & 1 & \dots & 1 \\ L_0 & L_1 & \dots & L_{n-1} \\ L_0^2 & L_1^2 & \dots & L_{n-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ L_0^{r-1} & L_1^{r-1} & \dots & L_{n-1}^{r-1} \end{bmatrix}, \quad D = \begin{bmatrix} D_0 & 0 & 0 & \dots & 0 \\ 0 & D_1 & 0 & \dots & 0 \\ 0 & 0 & D_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & D_{n-1} \end{bmatrix}.$$

Easily Decodable Codes

- N.B. The decoding algorithm may require a syndrome computed with such a special parity-check matrix H.
- Given a syndrome $c^T = Au^T$ computed with a different parity-check matrix A for the same code (hence $H = SA$ for some S), a decodable syndrome is obtained as $s^T = Sc^T = Hu^T$ with $S = HA^T(AA^T)^{-1}$.

Permuted Decoding

- **Problem:** Solve the GDP/SDP for a code $\mathcal{C}$ that is permutation equivalent to some efficiently decodable code $\mathcal{C}$'.
- Obvious resolution strategy: find the permutation and basis change between the codes, and use the $\mathcal{C}$' trapdoor to decode in $\mathcal{C}$.
- Conjectured to be "hard enough" for certain codes.

Shortened Decoding

- **Problem:** Solve the GDP/SDP for a code $\mathcal{C}$ that is permutation equivalent to some shortened (i.e. projection) subcode of some efficiently decodable code $\mathcal{C}$'.
- Obvious resolution strategy: find the permutation, basis change and shortening between the codes, and use the $\mathcal{C}$' trapdoor to decode in $\mathcal{C}$.
- Deciding whether a code is equivalent to a shortened code is NP-complete.

Cryptosystems Based on Syndrome Decoding

McEliece Cryptosystem

- Key generation:
  - Choose a uniformly random $[n, k]$ t-error correcting, efficiently decodable code Γ and a uniformly random permutation matrix $P \in (\mathbb{F}_q)^{k \times k}$, and compute a systematic generator matrix $G \in (\mathbb{F}_q)^{k \times n}$ for the equivalent code ΓP.
  - Set $K_{priv} = (\Gamma, P)$, $K_{pub} = (G, t)$.
- Encryption of a plaintext $m \in (\mathbb{F}_q)^k$:
  - Choose a uniformly random t-error vector $e \in (\mathbb{F}_q)^n$ and compute $c = mG + e \in (\mathbb{F}_q)^n$.
- Decryption of a ciphertext $c \in (\mathbb{F}_q)^n$:
  - Correct the errors in $c' = cP^{-1}$, i.e. find the t-error vector $e' = eP^{-1}$ s.t. $c' - e' \in \Gamma$, then recover m directly from $c - e \in \Gamma P$.

A Toy Example

- Let $n = 8$, $t = 1$, $k = 4$, and a code with the following systematic parity-check matrix H and generator matrix G:

$$H = \left[\begin{array}{cccc|cccc} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right], \quad G = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{array}\right].$$

- Encryption of the message $m = (1\ 1\ 0\ 0)$ with error vector $e = (0\ 0\ 1\ 0\ 0\ 0\ 0\ 0)$: $c = mG + e = (1\ 1\ 1\ 0\ 0\ 1\ 0\ 1)$.
- Syndrome computation $Hc^T = (1\ 1\ 1\ 1)^T$, error correction reveals e and yields $mG = c - e = (1\ 1\ 0\ 0\ 0\ 1\ 0\ 1)$.

Niederreiter Cryptosystem

- Key generation:
  - Choose a uniformly random $[n, k]$ t-error correcting, efficiently decodable code Γ and a uniformly random permutation matrix $P \in (\mathbb{F}_q)^{k \times k}$, and compute a systematic parity-check matrix $H \in (\mathbb{F}_q)^{r \times n}$ for the equivalent code ΓP.
  - Set $K_{priv} = (\Gamma, P)$, $K_{pub} = (H, t)$.
- Encryption of a plaintext $m \in (\mathbb{F}_q)^{\ell}$ with $\ell \le (n \text{ choose } t)$:
  - Represent m as a t-error vector $e \in (\mathbb{F}_q)^n$, and compute the syndrome $c^T = He^T \in (\mathbb{F}_q)^r$.
- Decryption of a ciphertext $c \in (\mathbb{F}_q)^r$:
  - Decode the syndrome $c^T = He^T = (HP^{-1})(Pe^T) = (HP^{-1})(eP^{-1})^T$ to the error vector $e' = eP^{-1}$ using the decoding algorithm for Γ, and obtain the plaintext m from $e = e'P$.

CFS Signatures

- Key generation:
  - Choose a uniformly random $[n, k]$ t-error correcting, efficiently decodable code Γ and a uniformly random permutation matrix $P \in (\mathbb{F}_2)^{k \times k}$, and compute a systematic parity-check matrix $H \in (\mathbb{F}_2)^{r \times n}$ for the equivalent code ΓP.
  - Choose a random oracle h: $\{0, 1\}^* \times \mathbb{N} \to (\mathbb{F}_2)^r$.
  - Set $K_{priv} = (\Gamma, P)$, $K_{pub} = (H, t)$.
- Signing a message m:
  - Find $i \in \mathbb{N}$ such that $s \leftarrow h(m, i)$ is a decodable syndrome of Γ, i.e. $s^T = He^T = (HP^{-1})(eP^{-1})^T$ for some t-error vector $eP^{-1} \in (\mathbb{F}_q)^n$.
  - Decode s^T to the error vector $e' = eP^{-1}$ using the decoding algorithm for Γ, and obtain $e \leftarrow e'P$. The signature is $(e, i) \in (\mathbb{F}_2)^n \times \mathbb{N}$.
- Verifying a signature (e, i):
  - Check that $w(e) \leq t$, and compute $c \leftarrow He^T$.
  - Accept the signature iff $c = h(m, i)$.

IND-CCA2 Security

- McEliece is not secure in the strong sense of indistinguishability under an adaptive chosen-ciphertext attack (e.g. $c = mG + e$ reveals all bits of m but t, at most).
- Solution: all-or-nothing transform (AONT), e.g. (McEliece-tailored) Fujisaki-Okamoto.

IND-CCA2 Security

- Random oracles
  - $\mathcal{R}$: $(\mathbb{F}_2)^k \to \{0, 1\}^*$.
  - $\mathcal{H}$: $(\mathbb{F}_2)^k \times \{0, 1\}^* \to \{0, \ldots, (n \text{ choose } t) - 1\}$, with output encoded as a vector in $(\mathbb{F}_2)^n$.
- Encryption of $m \in \{0, 1\}^*$:
  - $u \leftarrow$ random $(\mathbb{F}_2)^k$
  - $c \leftarrow \mathcal{R}(u) \oplus m$
  - $e \leftarrow \mathcal{H}(u, m)$
  - $z \leftarrow uG + e$
- The ciphertext is $(z, c) \in (\mathbb{F}_2)^n \times \{0, 1\}^*$.
- Decryption: find u and e from z, recover $m \leftarrow \mathcal{R}(u) \oplus c$, and accept iff $e = \mathcal{H}(u, m)$.

Summary

- Syndrome decoding based cryptosystems are simple and efficient.
- Security related to NP-complete and NP-hard problems (a suitable code may make this relation stronger).
- Strong notions of security are possible in the RO model using a suitable AONT.

Goppa Codes

Goppa Codes

- Let $g(x) = \sum_{i=0}^{t} g_i x^i$ be a monic ($g_t = 1$) polynomial in $\mathbb{F}_q[x]$ where $q = p^m$.
- 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. L is called the code support.
- 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 $S: (\mathbb{F}_p)^n \to \mathbb{F}_q[x]$:

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

- 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

- The syndrome can be written in parity-check matrix form as $H^* \in (\mathbb{F}_q)^{t \times n}$ or even $H \in (\mathbb{F}_p)^{mt \times n}$.

- Trace construction of the parity-check matrix H: write the $\mathbb{F}_p$ components of each $\mathbb{F}_q$ element (in a certain basis) from H^* on m successive rows of H.

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:

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

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

A Toy Example

- The toy example sets $m = 4$, $\mathbb{F}_{2^m} = \mathbb{F}_2[u]/(u^4 + u + 1)$, $n = 8$, $t = 1$, $k = n - mt = 4$, with generator polynomial $g(x) = x$ and support $L = (u^7, u^2, u^3, u^{10}, u^{13}, u^1, u^{11}, u^0)$.

- The parity-check matrix H^* (leading to the binary matrix H via the trace construction and systematic formatting) is

$$
\begin{aligned}
H^* &= TVD = \begin{bmatrix} u^8 & u^{13} & u^{12} & u^5 & u^2 & u^{14} & u^4 & u^0 \end{bmatrix}, \\
T &= \begin{bmatrix} 1 \end{bmatrix}, \\
V &= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \\
D &= \operatorname{diag}\begin{bmatrix} 1/g(L_0) & 1/g(L_1) & \dots & 1/g(L_7) \end{bmatrix}.
\end{aligned}
$$

Error Locator Polynomial

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

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

- Property: $\sigma(L_i) = 0 \Leftrightarrow e_i = 1$.
- For simplicity, assume binary fields (otherwise an error evaluator polynomial must be defined and computed as well).

Error Correction

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

Error Correction

- 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: $\frac{d}{dx}\sigma(x) = b(x)^2$. (N.B.: char 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 $\underbrace{a(x) = b(x)v(x)}_{\text{Extended Euclid!}}$ with $v(x) = \sqrt{x + \underbrace{1/s(x)}_{\text{Extended Euclid!}}} \bmod g(x)$.

A Toy Example

- The toy example sets $g(x) = x$, $L = (u^7, u^2, u^3, u^{10}, u^{13}, u^1, u^{11}, u^0)$, $c = (1\ 1\ 1\ 0\ 0\ 1\ 0\ 1)$, and $Hc^T = (1\ 1\ 1\ 1)^T$, so $s(\mathrm{x}) = u^3 + u^2 + u + 1 = u^{12}$.
- Hence $v(x) = (x + 1/s(x))^{1/2} \bmod g(x) = (x + u^3)^{1/2} \bmod x = (u^3)^{1/2} = u^9$.
- Extended Euclid starts with $a(x) = g(x) = x$ and $b(x) = 0$, and proceeds until $\deg(a) \le \lfloor t/2 \rfloor = 0$, $\deg(b) \le \lfloor (t-1)/2 \rfloor = 0$, with $a(x) = u^9$ and $b(x) = 1$.
- Thus $\sigma(x) = x + u^3$, which is zero for $x = u^3 = L_2$, and hence $e_2 = 1$ (i.e. c_2 is in error).

Summary

- Goppa codes are simple to construct and to decode.
- Binary irreducible Goppa codes have distance $2t + 1$. The best one gets for any other alternant code is distance $t + 1$.
- Cryptosystems on Goppa codes remain unbroken.

Problems and Challenges

Why Goppa?

- Most syndrome-based cryptosystems can be instantiated with general $[n, k]$-codes, but not all choices of code are secure.
  - Gabidulin, maximum rank distance (MRD), GRS, low-density parity-check (LDPC) and several other codes are all insecure.
- Goppa seems to be OK.
  - Complexity of distinguishing a permuted Goppa code from a random code of the same length and distance: $O(t\ n^{t-2} \log^2 n)$ [Sendrier 2000], or $O(2^n/t)$ in most cryptosystems, where $t = \Theta(n/\log n)$.
  - Few known vulnerabilities (e.g. generator polynomial defined over a proper subfield of the base field).

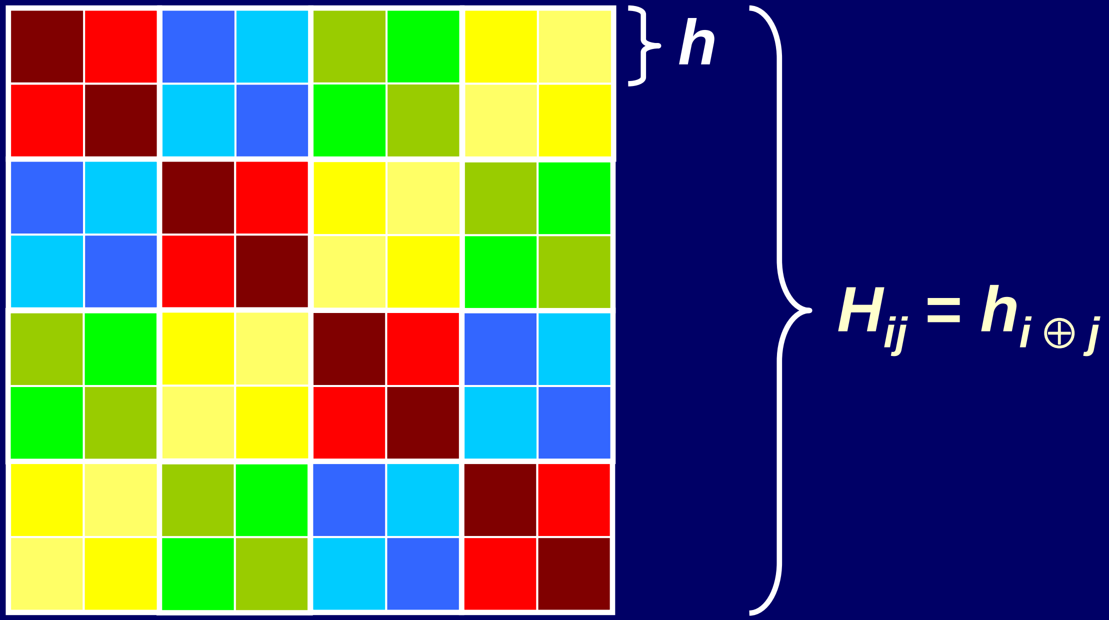
Choosing Parameters

- Original McEliece setting:
  - $m = 10$, $n = 2^m = 1024$ (hence L spans $\mathbb{F}_{2^m}$), $t = 50$, $k = n - mt = 524$, security ≈ 2^{54}, naïve key size = 65.5 KiB, key size = 32 KiB.
- Other choices [BLP 2008]:

security	n	t	k	m	naïve key size	key size
2^{80}	1632	33+1	1269	11	74–253 KiB	57 KiB
2^{128}	2960	56+1	2288	12	243–827 KiB	188 KiB
2^{192}	4624	95+2	3389	13	698–1913 KiB	511 KiB
2^{256}	6624	115+2	5129	13	1209–4147 KiB	937 KiB

Quasi-Dyadic Codes

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



Quasi-Dyadic Codes

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

Quasi-Dyadic Codes

- A Cauchy matrix is a matrix $C \in (\mathbb{F}_q)^{t \times n}$ where $C_{ij} = 1/(z_i - L_j)$ for vectors $z \in (\mathbb{F}_q)^t$ and $L \in (\mathbb{F}_q)^n$.
- Goppa codes admit a parity-check matrix in Cauchy form: just take z to be the roots of the generator polynomial, i.e. $g(x) = (x - z_0)\ldots(x - z_{t-1})$.
- **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 $H_{ij} = h_{i \oplus j} = 1/(z_i - L_j)$.

Quasi-Dyadic Codes

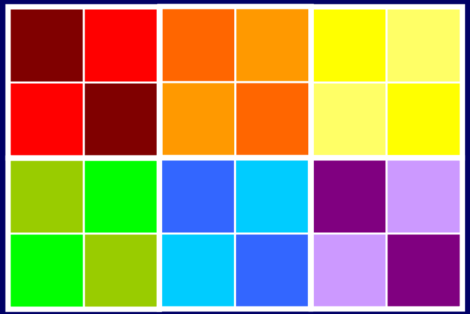
- Choose distinct h_0 and h_i with $i = 2^u$ for $0 \le u < \lceil \lg n \rceil$ uniformly at random from $\mathbb{F}_q$, then set

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

  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

- Sample parameters for practical security levels (private codes over $\mathbb{F}_{2^{16}}$).
- Still larger than RSA keys... but faster, and quantum-immune ☺

security	*n*	*t*	*k*	MB key size	BLP/MB
2^{80}	2304	64	1280	20480 bits	23
2^{128}	4096	128	2048	32768 bits	47
2^{192}	7168	256	3072	49152 bits	85
2^{256}	8192	256	4096	65536 bits	117

Further Issues

- One can do encryption, signatures, even identity-based identification using ECC (error-correcting codes, not elliptic curve cryptosystems).
- How do we get identity-based encryption? What about other protocols that are easy with pairings? N.B. Some functionality is possible with lattices – why not with ECC?

Appendix A

Hidden Subgroup Problem

- Let $\mathbb{G}$ be a group, $\mathbb{H} \subset \mathbb{G}$, and f a function on $\mathbb{G}$. We say that *f separates cosets* of $\mathbb{H}$ if $f(u) = f(v) \Leftrightarrow u\mathbb{H} = v\mathbb{H}$, $\forall u, v \in \mathbb{G}$.
- Hidden Subgroup Problem (HSP):
  - Let $\mathcal{A}$ be an oracle to compute a function that separates cosets of some subgroup $\mathbb{H} \subset \mathbb{G}$. Find a generating set for $\mathbb{H}$ using information gained from $\mathcal{A}$.
- Important special cases:
  - Abelian Hidden Subgroup Problem (AHSP)
  - Dihedral Hidden Subgroup Problem (DHSP)

Appendix B

Ranking and Unranking Permutations

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

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

- A *ranking function* is a mapping *rank*: $\mathcal{B}(n, t) \to \{1 \ldots r\}$ which associates a unique index in $\{1 \ldots r\}$ to each element in $\mathcal{B}(n, t)$. 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: $a_1a_2...a_n < b_1b_2...b_m$ in colex order iff $a_n...a_2a_1 < b_m...b_2b_1$ 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

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 $a_1\ a_2\ \ldots\ a_k$

Appendix C

Decoding a syndrome $s(x)$ for a binary Goppa code

$v(x) \leftarrow (x + 1/s(x))^{1/2} \bmod g(x)$ // extended Euclid!
$F \leftarrow v,\ G \leftarrow g,\ B \leftarrow 1,\ C \leftarrow 0,\ t \leftarrow \deg(g)$
while $(\deg(G) > \lfloor t/2 \rfloor)$ {
    $F \leftrightarrow G,\ B \leftrightarrow C$
    while $(\deg(F) \geq \deg(G))$ {
        $j \leftarrow \deg(F) - \deg(G),\ h \leftarrow F_{\deg(F)} / G_{\deg(G)}$
        $F \leftarrow F - h\, x^j\, G,\ B \leftarrow B - h\, x^j\, C$
    }
}
$\sigma(x) \leftarrow G(x)^2 + xC(x)^2$
return σ // error locator polynomial

Appendix D

Decoding Alternant Codes

- Similar to Patterson's algorithm for binary irreducible Goppa codes.
- Extended Euclid initialized with $s(x)$ instead of $v(x)$ and x^r instead of $g(x)$.
- $\sigma(x) = b(x)/b(0)$ (so that $\sigma(0) = 1$).
- N.B.: Patterson's algorithm works for binary reducible Goppa codes as long as the syndrome is invertible mod $g(x)$.