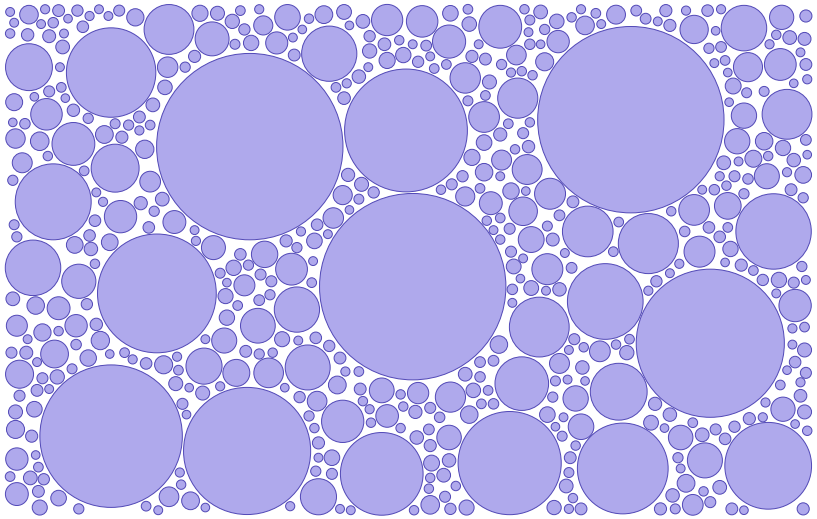
Introduction to (Hamming) code-based cryptography

Philippe Gaborit (XLIM, Université of Limoges, France)
Summer School, SAC 2026, Ottawa,
August 24, 2026



Summary

1. Background on coding theory
2. Post-quantum context
3. Difficulty of the syndrome decoding problem
    1. Theoretical complexity
    2. Practical complexity
4. Encryption/key exchange
    1. Original McEliece setting
    2. Quasi-cyclic McEliece setting
    3. Generalized McEliece / MDPC
    4. Aleknovich's random based setting/ Noisy DH and HQC
5. The NIST first competition
6. Signature with codes

Short history :

- **1940** : notion of error-correction introduced by R. Hamming from Bell Labs.
- **1948** : A mathematical theory of communication, by Claude Shannon manthematical bas of the theory and information theory

First applications : computer memories, deep space communications

General principle : adding redundancy bits to information bits

- **First example** : 1-error detection code : parity code, one add a pairty bits (used in DES)
$0\ 1\ 1\ 0\ 1\ 0 \rightarrow 0\ 1\ 1\ 0\ 1\ 0\ 1$

- allow to detect : **ONE** error : if the sum is not even it means there is one error somewhere
- **Second example** : 1-error correction code : 3-repetition code

$0110 \rightarrow 000\ 111\ 111\ 000$

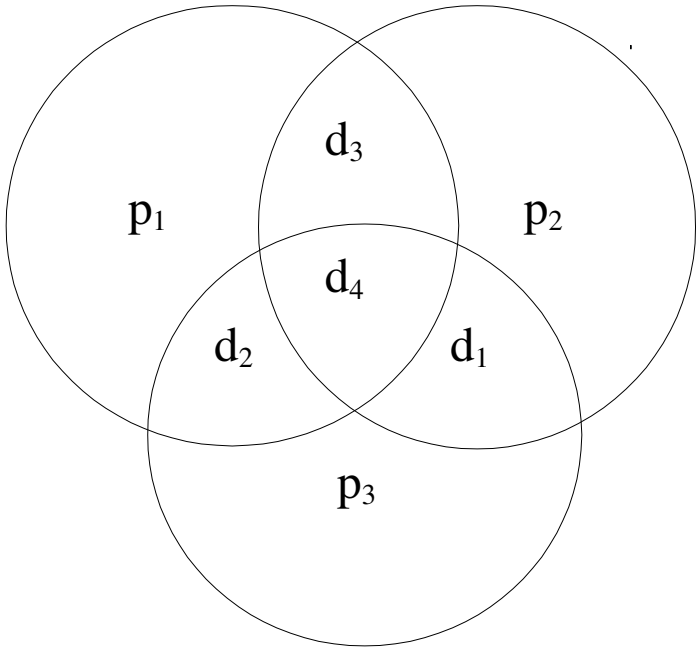
Majority decoding permits to decode one error

Third example : 1-error correction parity code

1	1	0
0	1	1
1	0	

Parity bit added for rows and columns $\rightarrow$ intersection of modified row and column permit to recover one error.

Fourth example : Hamming code



Information bits : d_1, d_2, d_3, d_4
Parity bits : p_1, p_2, p_3 sum in a circle is even.

$\Rightarrow$ permits to recover one erreur

- no parity in one circle $\rightarrow$ error $= p_i$ of this circle
- if no parity in 2 circles $\rightarrow$ error in their intersection (but not in 3 circle intersection) (d_2, d_3 ou d_4)
- if no parity in 3 circles : error $= d_1$

Definition

For $GF(q)$ a field, a $GF(q)$ linear error correcting code $C[n,k]$ of length n and dimension k is subspace of dimension k of $GF(q)^n$. A generator matrix of the code is a basis of subspace. For $x \in GF(q)^k$ an information vector, $c = xG \in GF(q)^n$ is an encoding of x.

Remark : up to permutation a generator matrix can be considered in systematic form [I A].

Generator matrix of the (binary) Hamming [7,4] code :

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Duality

Definition

For $x(x_1, .., x_n)$ and $y(y_1, .., y_n) \in F_q^n$, one considers the scalar product $:x.y = \sum_{i=1}^{n} x_i.y_i$.

Definition

Let C an $[n, k]_q$ code. Define the dual $C^\perp$ of C as $C^\perp = \{y \in \mathbb{F}_q^n | x.y = 0, \forall x \in C\}$.

Definition

Let C an $[n, k]_q$ code with generator matrix G. H is a parity check matrix of C if H is a generator matrix of the dual $C^\perp$. $c \in C$ iff $H.x^t = 0$

Definition

A parity check matrix of a generator matrix $(I_k A)$ is $(-A^t I_{n-k})$.

4 information bits and 1-error correction :

- Repetition code : $4 \rightarrow 12$
- Parity code : $4 \rightarrow 8$
- Hamming code : $4 \rightarrow 7$

Hamming code is in fact optimal.

Definition (Hamming weight)

Number of non null coordinates of x in $\mathbb{F}_q^n$, denoted $w_H(x)$.

example $x = (1, 0, 1, 0, 0, 1)$, $w_H(x) = 3$.

Definition (Hamming distance)

For x and y in $GF(q$, number of distinct coordinates between x and y, $d_H(x, y) = w_H(x - y)$.

Definition (Minimum distance)

For a linear code C, its minimum $d(C)$ is the smallest weight of non null element of the code. $d(C) = min_{x\in,x\neq 0} w_H(x)$. One denotes by C[n,k,d], a code of length n, dimension k and minimum distance d.

Definition (Maximum likelyhood decoding)

Given a [n,k,d] code and et un mot $y \in \mathbb{F}_q^n$, a maximum likelyhood decoding of y consists in associating to y the nearest codeword of for Hamming distance.

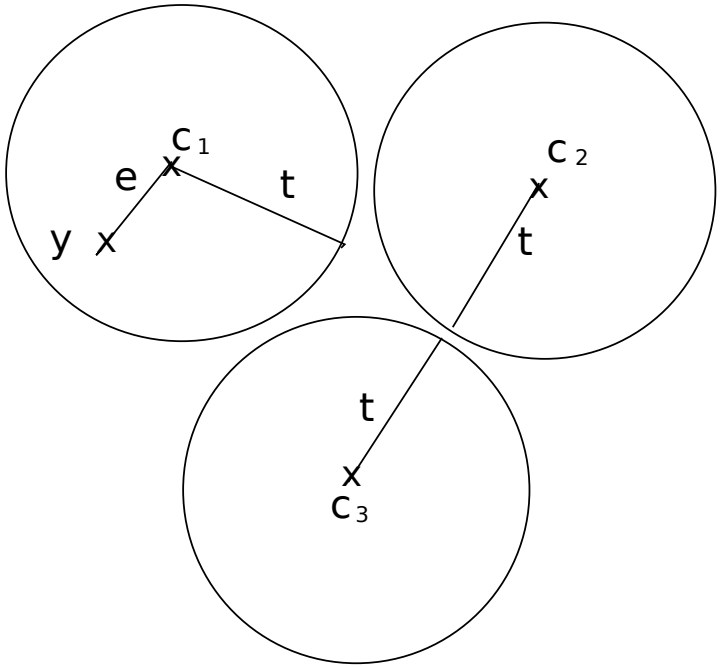
$\texttt{Decode}(y,) = x \quad \text{such that} \quad d_H(x,y) = min_{c \in} d_H(c,y).$

Theorem

For an $[n,k,d]_q$; it is possible to decode in a unique way by maximum likelyhood decoding up to errors of weight $t = \lfloor \frac{d-1}{2} \rfloor$.

proof : if there existed two codewords c_1 and c_2 with a distance to y smaller than t then $c_1 - c_2 \in C$ and $d(c_1,c_2) \leq d(c_1,y) + d(c_2,y) < d$, not possible.

t=[(d-1)/2]



Decoding problem

Given c=xG a codeword, an error e of weight t and a received vector y=c+e :
Decoding problem Recover c from y, $y \rightarrow (c, e)$

Syndrome decoding problem : Recover e from syndrome $s = H.e^t$

Since $y = xG + e$ and $H.y^t = H.e^t = s$ both problem are equivalent

General problematic of error correcting codes

1) Construct "good codes" with good ratio k/n which corrects many errors
$\rightarrow$ antinomic conditions !

2) Find good decoding algorithm (typically in polynomial time ($O(n^2)$ or $O(n^3)$) better than trivial syndrome decoding algorithm

- often codes are found separately from decoding
- over GF(2) for k/n very small $\rightarrow$ error of rate up to $1/2$
- over GF(q) for k/n very small $\rightarrow$ error of rate up tending to 1

Examples of families f codes

- 1940 codes invented by R. Hamming at Bell Labs.
- 1948 A mathematical theory of communication, by C. Shannon
- 1949 Golay codes (in a sequence contexte)
- 1954 Reed-Muller codes
- 1957 Cyclic codes
- 1959 BCH codes
- 1960 Reed-Solomon codes
- 1962 LDPC codes by Gallagher
- 1970 Goppa codes
- 1978 MacEliece crypto
- 1982 Algebraic Geometry codes
- 1993 Turbo codes
- 1997 renewal on LDPC codes

The case of random codes

- Gilbert-Varsahmov bound d_{GV} : average minimum weight of a random [n,k] code
- syndromes $H.e^t$ behave as random elements of $GF(2)^{n-k}$:

$$\binom{n}{d_{GV}} \approx 2^{n-k}, \text{with} \binom{n}{d} \approx 2^{nH_2(\frac{d}{n})},$$

for $H_2(x) = -xlog_2(x) - (1-x)log_2(1-x)$

$$d_{GV} \approx nH_2^{-1}(1 - \frac{k}{n}).$$

For $\frac{k}{n} = \frac{1}{2}, d_{GV} \approx \frac{n}{9}$

Use of cyclicity

$\mathcal{R} = \mathbb{F}_q[X]/(X^n - 1)$
For any two elements $x, y \in \mathcal{R}$, their product, defined as the usual product of two polynomials modulo $X^n - 1$, is as follows :
$x \cdot y = z \in \mathcal{R}$ with

$$z_k = \sum_{i+j \equiv k \mod n} x_i y_j, \text{ for } i, j, k \in \{0, \ldots, n-1\}.$$

Notice that as the product of two elements over the *commutative* ring $\mathcal{R}$, we have $x \cdot y = y \cdot x$.

Polynomial notation : $x(x_0, .., x_{n-1}) \rightarrow p(x) = \sum_{i=0}^{n-1} x_i X^i$

$$x.y = p(x).p(y) mod(X^{n-1} - 1)$$

Use of cyclicity

Definition (Circulant Matrix)

Let $x = (x_0, \ldots, x_{n-1}) \in \mathbb{F}_q^n$. The *circulant matrix* induced by x is defined and denoted as follows :

$$\mathbf{rot}(x) = \begin{pmatrix} x_0 & x_1 & \cdots & x_{n-1} \\ x_{n-1} & x_0 & \cdots & x_{n-2} \\ \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & \cdots & x_0 \end{pmatrix} \in \mathbb{F}_q^{n \times n} \quad (1)$$

As a consequence, it is easy to see that the product of any two elements $x, y \in \mathcal{R}$ can be expressed as a usual vector-matrix (or matrix-vector) product using the $\mathbf{rot}(\cdot)$ operator as

$$x \cdot y = x \times \mathbf{rot}(y) = \left(\mathbf{rot}(x)^\top \times y^\top\right)^\top = y \times \mathbf{rot}(x) = y \cdot x. \quad (2)$$

$\rightarrow$ approach can be extended to ideal rings with different type of polynomials ($X^n + 1$ for instance or block of circulant matrices)

Post-quantum (or quantum-resistant) context

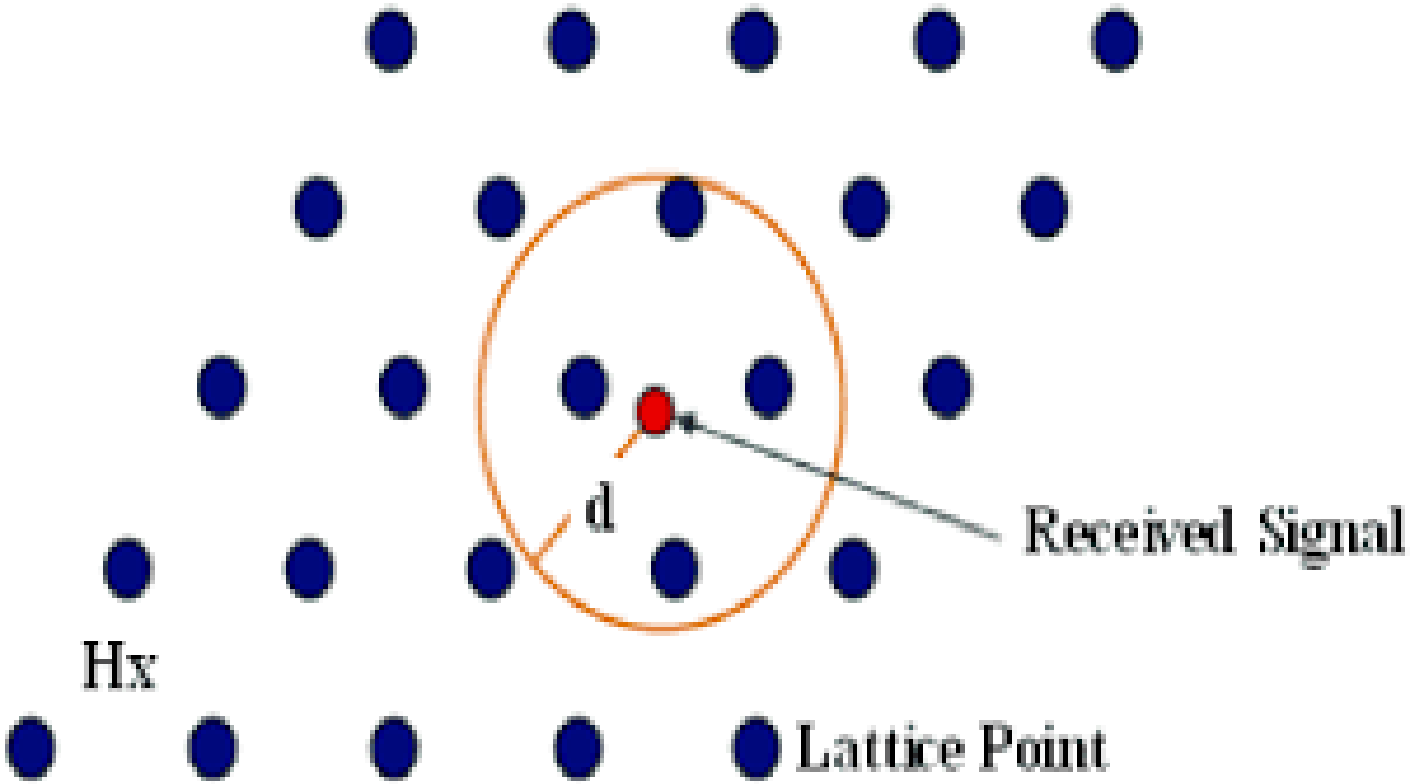
General problems

Cryptography needs different difficult problems

- factorization
- discrete log
- SVP for lattices
- syndrome decoding problem

For code-based cryptography, the security of cryptosystems is usually related to the problem of syndrome decoding for a special metric.

Distance-based cryptography,general problem : the decoding problem



Decoding problem :

G a random $k \times n$ matrix over Z/qZ, $x \in (Z/qZ)^k$

Encoding : x rightarrow y= x G + e

for e a small weight vector in $(Z/qZ)^n$

y : received vector

Decoding problem equivalent to syndrome decoding problem :

Given $s = H.e^t$ (for e small) recover e.

(for H a dual matrix compute $H.y^t = H(xG + e)^t = H.e^t$.)

Problem proven NPC over GF(2) with Hamming metric in '78 (Berlekamp,McEliece, VanTilborg) then later over other type of metrics.

- related problem Learning With Errors (LWE) and Learning Parity with Noise (LPN), essentially same problem with e following a given distribution.

Relation with linear algebra

Consider the following linear system : H a random $(n-k) \times n$ over $Z/2Z$

Knowing $s \in (Z/2Z)^{n-k}$ is it possible to find an $x \in (Z/2Z)^n$ such that $H.x^t = s$?

EASY ! ! !

- Fix $n-k$ columns of H , one gets $(n-k) \times (n-k)$ submatrix A of H
- A is invertible with a good probability : fix $x' = A^{-1}s$ and $x = 0..00x'00..00$.

How is it possible to turn this problem into a difficult problem ?

(1) add a constraint to x : x of small weight for a particular metric

if n=1000, k=500, one can fix weight(x) = 110

$$\left(\begin{array}{cccccccccccc|c} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & \\ \hline 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & \end{array}\right) \times X = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$X = (1000000000100)$$

More generally over Z/qZ :

$$\left(\begin{array}{cccccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ \hline -2 & 7 & -4 & 1 & 31 & -9 & -4 & 3 & 12 & 18 & -23 & 1 \\ -11 & 1 & 5 & 11 & 7 & -12 & 7 & 1 & 31 & -21 & 13 & -3 \\ -1 & 30 & 5 & 23 & 7 & 8 & -4 & 17 & 11 & 0 & -24 & 1 \\ 8 & -4 & 1 & -1 & 9 & 10 & 17 & -21 & 4 & 8 & 13 & -11 \end{array}\right) \times X = \left(\begin{array}{c} -22 \\ -25 \\ -11 \\ 16 \end{array}\right)$$

$$X = (1, -1, 0, 1, -1, 0, 1, -1, 0, 0, -1, 1)$$

Words of the code $\mathcal{C}$ are n-uplets with coordinates in $GF(q^m)$.

$$v = (v_1, \ldots, v_n)$$

with $v_j \in GF(q^m)$.
Any coordinate $v_j = \sum_{i=1}^{m} v_{ij} b_i$ with $v_{ij} \in GF(q)$.

$$v(v_1, ..., v_n) \rightarrow V = \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{pmatrix}$$

- metric = Hamming distance (over $(Z/2Z)^n$) $\Rightarrow$ **Hamming based cryptography (code based cryptography)**
  $(0,0,1,0,0,1,0,0,1) \rightarrow$ Hamming weight 3
- metric = Euclidean norm (over $(Z/pZ)^n$) $\Rightarrow$ **lattice based cryptography**
  $(2,1,-1,3,-2) \rightarrow$ weight
  $=\sqrt{2^2+1^2+(-1)^2+3^2+(-2)^2}=\sqrt{19}$
- metric = Rank metric $\Rightarrow$ **rank based cryptography**

Comparison between metrics

problem : H a $(n-k) \times n$ matrix over K$= Z/qZ$ or $GF(q)$ or $GF(q^m)$ for $s \in K^n$ find a solution to

$$H.x^t = s$$

- NP-hard problem for any metric (reduction depending on the metric)
- the complexity has a max value depending on the weight of x corresponding to the case where the number of words of weight w $\sim |K|^{n-k}$

Comparison between metrics

- **hardest weight value** = Gilbert-Varhsmov bound (GV for Hamming) or Rank GV for Rank also called Gauss heuristics for Euclidean distance.
- notion of value of weight for which solving the problem is easy :

Hamming : (n-k)/2
Rank : (n-k)m/n
Euclidean : $O(2^n)$

ratio : **ceiling** = $\frac{\textbf{easy weight}}{\textbf{hardest weight}}$

Hamming : log(n) (for k $\sim$ n, else for k=n/2, 9/4)
Rank : 2
Euclidean : potentially exponential, in practice poly(n)

$\rightarrow$ a lot of margin for lattices..... better for advanced scheme (homomorphic scheme for instance) and signatures.

(2) consider rather a multivariable non linear system : quadratic, cubic etc...

$\Rightarrow$ Mutivariate cryptography

Difficult code-based problems used for cryptography - theoretical approach

Definition (Syndrome Decoding problem)

Let H be a random matrix $(n-k) \times n$ over F_2^n, let s a word of F_2^{n-k} and let w a postive integer.
Question : does there exist a word x of F_2^n of weight $\leq w$ such that $Hx^t = s$?

NP-complete problem Berlekamp, McEliece and van Tilborg '78
proof : Reduce 3D matching problem to SD problem.

a variation :

Definition (Minimum Weight problem)

Let H be a random matrix $(n-k) \times n$ over F_2^n, let w a postive integer.
Question : does-there exist a word x of F_2^n of weight $\leq w$ such that $Hx^t = 0$?

Clearly SD $\Rightarrow$ MinWeight, reciprocally ? ? ?, easy to prove that previous problem equivalent to :

Definition (Minimum weight problem)

Given a random generator matrix G, and let C be the associated code.
Question : What is the minimum weight of C ?

proven NP-complete by Vardy '97 (more difficult to prove)

other problems :

Definition (Decisional Syndrome Decoding problem)

Let H be a random matrix $(n-k) \times n$ over F_2^n, let s a word of F_2^k and let w be a postive integer.
Question : Given x of weight w, is it possible to distinguish between Hx^t and r random in F_q^{n-k} ?

easy to prove equivalent to SD for a perfect adversary with 0/1 advantage (rather than a proba bias)
- more difficult to prove if only one small advantage : GoldReich-Levin theorem but eventually for SD : the decisional version and the search version are equivalent.

CONCLUSION

1. Previous results show that the problem of decoding random codes is hard (ie. it is not believed a polynomial algo will be found to solve them generically)
2. for codes even if no theoretical results are known, it is believed the problems remains hard "on the average".

Practical complexity of the SD problem

General complexity of the syndrome decoding problem

Given random H($n-k \times n$) and $s = Hx^t$ for x of weight t, what is the cost of recovering x ? ? ? Very studied problem, several variables n,k,t, case of several vectors to decode, "unicity" max for GV. Comparison for this case.
Complexity has exponent in $2^{\alpha n}$ (up to log and polynomial factors) for the worst case (in t and n) (hence $R = \frac{1}{2}, t = GV(\approx \frac{n}{9})$

Name	date	α
Exhaustive search		0.386
Prange	1962	0.1207
Stern	1988	0.1164
Dumer	1991	0.1162
May, Meurer, Thomas	2011	0.1114
Becker,Joux, May, Meurer	2012	0.1019
May, Ozerov	2015	0.1019
Both , May	2017	0.0953
Both, May	2018	0.0885

Rate smaller than 0.3 : Statiscal Decoding 2.0 and 3.0 (2022,2024) K Carrier, T Debris-Alazard, C Meyer-Hilfiger, JP Tillich

Main tool : Information Set Decoding (Prange 1962)

Classical approach (and name) Let C $[n, k]$ code, y=xG+e of weight t (small)
Choose a random set of k colonnes, an error e of weight t is said decodable whenever its support does not meet the k random columns.

Dual approach with a parity check matrix : $s = H.e^t$, choose (n-k) columns, hope they contain the support of the error, solve the system from the associated syndrom, distinguisher = the weight of the solution.

Prange Algorithm '62 : $H.e^t = s, H : (n-k) \times n$, weight(e)= t

1) **pick n-k random columns** , wlog the first n-k columns,
$e = (e_1, e_2)$

$$(H'|H'').(e_1, e_2)^t = s$$

2) **multiply by** H'^{-1}

$$(I|H'^{-1}H'').(e_1, e_2)^t = H'^{-1}.s$$

3) **weight condition**
- **if** $weight(H'^{-1}.s) = t$: take $e_1 = H'^{-1}.s$ and $e_2 = 0$ then $(e_1, 0)$ is a solution of $H.e^t = s$ and the problem is solved

- **if** $weight(H'^{-1}.s) > t$: return to 1)

Probability of good guessing : $P_{dec} = \frac{\#\ good\ choices\ of\ columns}{\#\ choices} = \frac{\binom{n-t}{n-k-t}}{\binom{n}{n-k}} = \frac{\binom{n-t}{k}}{\binom{n}{k}}$,

hence $P_{dec} = O(1).2^{-nH_2(t/n)-(n-k)H_2(t/(n-k))}$, for
$H_2(x) = -xlog_2(x) - (1-x)log_2(1-x)$.

Then complexity : finding ONE word of weight t :

$$WF = \frac{P(k)}{P_{dec}},$$

with $P(k)$ cost of a matrix inversion. $P(k)$ of order $O(k^3)$, (possible to improve).

Improvements

1. No memory needed for Prange
2. Lee-Brickell '88 (A,I) with x of the form (p,w-p) exhaustion on p (no memory)
3. Stern '89/Dumer '91 (H_1, H_2, I) with x (p/2,p/2,w-p) Meet in the middle (memory)
4. other improvements by representation (large memory needed, high constant term (Becker, Joux, May Meurer (2012))
5. improvement of nearest neighbor search MO 15 (large memory needed)

All improvement based on Prange + improvement of meet in the middle
In practice best attacks for Minimum weight or decisonal SD = attacking SD problem.
Estimator : https ://estimators.crypto.tii.ae/

Special case t=o(n)

In that case when $t = o(n)$ best complexity (CantoTorres-Sendrier 2016)

$$2^{-log(1-R)t(1+o(1))}$$

All previous attacks give asymptocically the same exponential complexity (see curves in k/n and complexity).

The asymptotic part of this case has not been improved in fact since Prange (more than 60 years !)

Quantum attacks

Very few "main" quantum algorithms

1. '94 Shor factorization (fast computation of the order of an element of a group) log(n) improvement (makes factorization, DL, EC-DL, polynomial)
2. '96 Grover search permits to search a list with an improvement in $\sqrt{n}$. In practice it means dividing the exponent by a factor 2.
3. Quantum walks, same type of square root gain.

- evolution of quantum computers from 10 qubits to ? ? ? ?, interest ? ?, logical qbits IBM ? ? ?

- in practice (Bernstein '08) Grover algorithm permits to divide the Prange exponent by 2.

- other improvements use birthday paradox hence, smaller gain, in practice only minor and for quantum attacks best complexity is very close from quantum-Prange

Original McEliece setting for code-based cryptography

McEliece Cryptosystem '78

Comes right after the Merkle-Hellman system (only a report at JLC)

1. **Key generation**
   1. Private key : C an $[n, k, d]$ code correcting t errors, with generator matrix $k \times n$ G, matrix $k \times k$ S invertible, and $n \times n$ P permutation matrix.
   2. Public key : $G' = SGP$
2. **Encryption** : $x \rightarrow xG' + e$ with $w_H(e) \leq t$.
3. **Decryption** : $y \rightarrow \Phi_C(yP^{-1})S^{-1}$, with $\Phi_C(z)$ a decoding algo up to distance t.

why it works : $yP^{-1} = (xS)G + eP^{-1}$.

Proposed codes : Goppa codes [1024,524,101], t=50.

Complexity :
Encryption : $O(n^2)$
Decryption : $O(n^2)$
Key size : $k(n-k)$ (matrix G' in systematic form)
Message length : n
Transmission rate : k/n
$\rightarrow$ system very fast but *very* large size of key.
500.000 bits for the original system !

Very close in the spirit to Merkle-Hellman cryptosystem : masking of an easy instance

- easy instance : super increasing sequence $\rightarrow$ decodable code

- masking : multiplication by p modulo N $\rightarrow$ permutation masking

Niederreiter scheme '86 (dual scheme of McEliece)

1. **Key generation**
   1. Private key : C an $[n, k, d]$ code correcting t errors, with parity check matrix H $(n-k) \times n$, matrix $(n-k) \times (n-k)$ invertible, and $n \times n$ P permutation matrix.
   2. Public key : $H' = SHP$
2. **Encryption** : message $\rightarrow$ error e of weight $w_H(e) \leq t$, ciphertext $y = H.e^t$.
3. **Decryption** : decode $S^{-1}y$ in z, then $zP^{-1} = x$ and x gives m.

Historically : propose to use Reed-Solomon codes.

Comparison McEliece / Niederreiter

- Equivalent security / size of key
- **advantage** : Niederreiter scheme has smaller ciphertext size
- **disadvantage** : the message has to be turned into a codeword of fixed weight

$\rightarrow$ can really be a problem ! !

Security

2 types of generic attacks :

- **Message attack**

In that case the attacker tries to decode directly the ciphertext. Under the undistinguishability of the public, it corresponds to decoding a random code. See previous attacks.

$\rightarrow$ Exponential complexity of the best known algorithms.

Originel cryptosystem of McEliece originel broken theoretically in '98 par Canteaut, in practice only in 2008 by Bernstein-Lange..

Notice that the security of the cryptosystem does not rely directly on SD, and hence is not clearly based on a NPC problem.

Structural attack

For this attack the attacker wants to retrieve the structure of the private key from the public key. A way for instance to do that is recovering the permutation matrix.

Definition (Code equivalence problem)

Let C be a code of length n and P be a permutation on n columns.
Question : Is possible to recover the permutation P from (C,CP) ?

Decision version : Given C, CP and D(random), distinguish between CP and D.

Problem a priori not NPC, similar to graph equivalence, smaller class of complexity.

1. Leon algorithm : compute the set of ALL small weight vectors of C and CP then use a back track search algorithm on possible permutations ; exponential complexity for random codes
2. Support Splitting algorithm (Sendrier '97), same that Leon but starting from the hull ($C \cap C^{\perp}$), (works even if the dimension of the hull is zero - just truncate on some positions) : complexity in $O(2^{dim(C \cap C^{\perp})})$.

- Random codes have a small hull in general (Sendrier '97) : dimension 0.75... the algo does not work for large hulls : Reed-Muller or BCH codes for instance.

- Example of problem, easy on the average , with some hard instances, general complexity unknown.

$\rightarrow$ **To resist :** either a large hull (like BCH but not necessarilry sufficient), either starting from a very large set of "independent" codes like Goppa codes or alternant codes.

Examples

The McEliece cryptosystem can be used with "any code", it is more a setting than a cryptosystem. It mainly relies on the **indistinguishability** of the public key. Not ind-cpa (one can test a message for instance, non randomized - as textbook RSA)
Historically, semantic security was less explored by people in code-based crypto.

Sendrier proposes in 2002 the "Goppa indistingushability" problem (broken in 2012 by Fauger, Otmani Tillich et al. for some type of parameters).
- very recently :
(Ghoshal, Ishai, Jain, Sun) :Quasipolynomial Cryptanalysis of the McEliece Cryptosystem (or : PIR Meets McEliece)
https ://eprint.iacr.org/2026/1630, propose quasi-polynomial attack for distinguishing (and an heuricitic for the search problem)

Examples of systems

1978 McEliece : binary Goppa
1986 Niederreiter variations : GRS codes
1991 Gabidulin, Paramonov, Tretjakov : Gabidulin codes
1994 Sidelnikov : Reed-Muller codes
1996 Janwa-Moreno : algebraic geometry codes
199* many propositions with LDPC codes
2003 Alekhnovich : system Alekhnovich
2005 Berger-Loidreau : sub-codes of GRS
2006 Wieschebrink, GRS codes + random columns in gen. matrix
2008 Baldi et al. : MDPC codes based on LDPC codes
2010 Bernstein, Lange, Peters : savage Goppa codes
2012 Misoczki-Tillich-Barreto-Sendrier : codes MDPC
2012 Londahl-Johansson : convolutives codes
2013 Gaborit, Murat, Ruatta, Zemor : LRPC codes
2014 Shrestha, Kim : polar codes
2014 Hooshmand, Shooshtari, Eghlidos, Aref : sub codes of polar codes

1978 McEliece : binary Goppa
1986 Niederreiter variations : GRS codes
1991 Gabidulin, Paramonov, Tretjakov : Gabidulin codes
1994 Sidelnikov : Reed-Muller codes
1996 Janwa-Moreno : algebraic geometry codes
199* many propositions with LDPC codes
2003 Alekhnovich : system Alekhnovich
2005 Berger-Loidreau : sub-codes of GRS
2006 Wieschebrink, GRS codes + random columns in gen. matrix
2008 Baldi et. al : MDPC codes based on LDPC codes
2010 Bernstein, Lange, Peters : savage Goppa codes
2012 Misoczki-Tillich-Barreto-Sendrier : codes MDPC
2012 Londahl-Johansson : convolutives codes
2013 Gaborit, Murat, Ruatta, Zemor : LRPC codes
2014 Shrestha, Kim : polar codes
2014 Hooshmand, Shooshtari, Eghlidos, Aref : sub codes of polar codes

Example of attacks

Main weakness of McEliece setting : the indistinguashibility of the public key ? ?
How to prove that it is difficult to recover the structure from the knowledge of the public key ?

- **Sidelnikov-Shestakov attack '92 on RS codes :** from a masked codes (GRS + permutations : it is possible to recover the structure) in POLYNOMIAL time.

$\rightarrow$ difficulty to hide RS, very strong polynomial structure

- **attack on the dual :** product codes by Sendrier / LDPC

Example of attacks

- **square attack :** remark that if $c(f)$ and $c(g)$ are words of a RS(n,k), then f and g are polynomials of degree less than $n-1$ then componentwise vector $c(f) * c(g)$ is of degree up to $2k-2$.

Definition (Square attack - CGOPT '15)

Let C=RS(n,k), then the square code $C * C$ is a RS code of dimension at most $2k-2$. Hence for k small (typically less than $\sqrt{n}$) there is possibility to distinguish between $C * C$ where C is a RS code or C is random.

example : Wieschbrink's system permuted (RS + Random)

Baldi et al. system : multiplication by a non permutation matrix : weight between 1 et 2 per columns, attack in some cases (PKC 2015)

Advantage of RS codes $\rightarrow$ smaller key size than McELiece (factor 10 say) but more difficult to hide.

RS : better key size but hard to mask,
alternant codes : better masking when the order decreases but less good codes (except Goppa codes)

$\rightarrow$ in practice Goppa of order 2 can also be attacked, for binary Goppa codes, no attacks known, systems seems resistant but always a doubt. (BCH not attacked either).

In general all other families of codes are attacked at divers level : often completely broken (polynomial) either partially attacked (RM, AG codes, convolutive, polar etc...) but then there is doubt... that the attacked can be pushed farther.....

In general it is/was wildly thought that McEliece-(binary)-Goppa is safe. (square attack does not work with these parameters).

However recently new approaches :
-Magali Bardet, Rocco Mora, Jean-Pierre Tillich : Polynomial Time Key-Recovery Attack on High Rate Random Alternant Codes (2024)
- Alain Couvreur, Rocco Mora, Jean-Pierre Tillich : A New Approach Based on Quadratic Forms to Attack the McEliece Cryptosystem (2023).
- Hugues Randriambololona, The syzygy distinguisher (2024)
and very recently :
Ghoshal, Ishai, Jain, Sun :Quasipolynomial Cryptanalysis of the McEliece Cryptosystem (or : PIR Meets McEliece)
https ://eprint.iacr.org/2026/1630, propose quasi-polynomial attack for distinguishing (and an heuristic for the search problem)

Quasi-cyclic Mc Eliece settings

How to shorten the size of the keys (Gaborit '05)

Main idea : use a compact representation of a code and a masking compatible with this compact representation.

Consider generator matrix of a quasi-cyclic code :

Proposition

Let C a $[n, k]$ quasi-cyclic code of order s with $n = rs$ then ther exists a matrix $k' \times n$ G (with $k' \geq k$), which generated C of the form :

$$G = \begin{pmatrix} A_1 & A_2 & A_3 & \cdots & A_r \\ A_r & A_1 & A_2 & \cdots & A_{r-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_2 & A_3 & A_4 & \cdots & A_1 \end{pmatrix},$$

with A_i $\frac{k'}{r} \times s$ matrices.

$$\Pi = \begin{pmatrix} \pi & 0 & \cdots & 0 \\ 0 & \pi & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \pi \end{pmatrix},$$

for S_{QC} an random QC matrix, then masked matrix :

$$G' = S_{QC} \times \begin{pmatrix} A_1\pi & A_2\pi & A_3\pi & \cdots & A_r\pi \\ A_r\pi & A_1\pi & A_2\pi & \cdots & A_{r-1}\pi \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_2\pi & A_3\pi & A_4\pi & \cdots & A_1\pi \end{pmatrix},$$

$\Rightarrow$ Smaller key parameters, the general McEliece scheme remains the same.

Choice of codes

One wants : codes with : good decoding, and quasi-cyclic, with sufficiently many codes.

- Gaborit '05 : quasi-cyclic subcodes of BCH narrow-sense primitive codes. (bad choice of codes : polynomially broken 2008 Otmani-Tillich).

- T.P. Berger, Pierre-Louis Cayrel, Philippe Gaborit, Ayoub Otmani (2009) : QC Alternant codes : parameters too optimistic (parameters broken 2010 Faugere et al.)

- Misoczki-Barreto 2010 : Dyadic Goppa structure (some unbroken)

- DAGS : optimistic dyadic (NIST 2017) - parameters broken and broken again (Bareli-Couvreur 2018)

-BigQuake : QC Goppa (NIST Couvreur et al. 2017), gain factor 7-10, unbroken.

$\rightarrow$ Approach remains interesting if one is not too greedy, needs more algebraic attacks analysis.

Generalized McEliece setting and the particular case of LDPC based approach and MDPC codes

Generalized McEliece Scheme

1. **Key generation**
   1. Private key : C an $[n, k, d]$ code correcting t errors, with generator matrix $k \times n$ G, TrapDoor() : a trapdoor transformation.
   2. Public key : $G' = Trapdoor(G)$
2. **Encryption** : $x \rightarrow xG' + e$ with $w_H(e) \leq t'$.
3. **Decryption** : $y \rightarrow \Phi_C(TrapDoor^{-1}(y))$, with $\Phi_C(z)$ a decoding algo up to distance t.

Why it works : $TrapDoor^{-1}(y) = x'G + e'$, weight(e') $\leq t$

Example : G'=Trapdoor(G)=SGP'^{-1}, P' : weight 2 matrix
$TrapDoor^{-1}y = yP' = (xS)G + eP', weight(e' = eP) \leq 2t'$

LDPC/MDPC codes

LDPC : Gallagher 1963
LDPC codes = Low Density Parity Check codes defined from a sparse parity check matrix

General idea : use the fact that the parity check matrix is sparse to decode.

The syndrome is a linear combination of (very) small weight columns, hence it is possible to link probabilistically a given column with small weight to the syndrome by counting the number of intersected '1'.

BitFlip algorithm One step : for all columns : one counts the number of intersected bit for each column and then one may flip bits or not, the process is done iteratively with a certain number of steps.

$\rightarrow$ permits to obtain an efficient iterative decoding.

example :

1111	0000	0000	0000
0000	1111	0000	0000
0000	0000	1111	0000
0000	0000	0000	1111
1000	1000	1000	1000
0100	0100	0100	0100
0010	0010	0010	0010
0001	0001	0001	0001
1000	0100	0010	0001
0100	0010	0001	1000
0010	0001	1000	0100
0001	1000	0100	0010

(16,3,4) LDPC code : length 16, 3 1's per column, 4 per row.
Corresponds to a [16,6,6] code.

In practice imagine length 100.000 and 3 '1' per column.

Low DPC codes (LDPC) : constant weight $\Rightarrow$ decoding O(n) (**Unsafe for crypto**)

Moderate DPC codes (MDPC) : weight $O(\sqrt{n}) \Rightarrow$ decoding $O(\sqrt{n})$ (**ok for crypto**)

LDPC/MDPC approach

- Many systems in the '90 '00 but problem of small vectors in the dual.

- '08 Baldi et al. consider an Quasi-Cyclic LDPC matrix masked with a matrix which multiplies the weight.
$\rightarrow$ in practice : LDPC decoding of (weight of error) x constant
good idea, but decoding not optimal and ultimately a small structural attack (Apon et al. CRYPTO 2020, LEDACRYPT) .

- Mizocki,Tillich, Barreto and Sendrier '13 QC-MDPC (Quasi-Cyclic Moderate Density Parity Check codes).

$\rightarrow$ same type of previous ideas but with an optimal decoding and Quasi-Cyclicity.

QC-MDPC decoding analysis

Very complex to analyze.

Remarks :
- the MDPC bit-flip algorithm is probabilistic

- mitigation between : very small weight vectors per row : constant (3,6) which can decode almost up to Shannon limits O(n) and this case : the small weight is in $O(\sqrt{n})$, clearly higher, hence can decode but only in $O(\sqrt{n})$

- possibility to simulate the error decoding probability , but difficult to have theoretical bounds, in practice works well.

- many possibilities to optimize the decoding : adaptable threshold, fixed time decoder

- many papers on the subject, see : Nicolas Sendrier, Valentin Vasseur, , eventually :

Sarah Arpin, Jun Bo Lau, Antoine Mesnard, Ray A. Perlner, Angela Robinson, Jean-Pierre Tillich, Valentin Vasseur : Error Floor Prediction with Markov Models for QC-MDPC Codes. CRYPTO 2025
propose a precise analysis.

QC-MDPC Cryptosystem / BIKE

1. **Key generation**
   1. **Secret key** x,y small weight random vectors of weight of order $\sqrt{n}$
   2. **Public key** $h = x.y^{-1}$ (considered in polynomial form modulo $X^n - 1$)
2. **Encryption** $m \rightarrow e = (e_1, e_2)$ of small weight $\sim O(\sqrt{n})$.
   $c = (1, h).(e_1, e_2)^t = e_1 + h.e_2$
3. **Decryption** compute $x.c = x.(e_1 + e_2.h) = x.e_1 + y.e_2 = (x, y).(e_1, e_2)^t$ which can be decoded as a MDPC code.

$\rightarrow$ the polynomial computations correspond to manipulating double-circulant matrices, the ciphertext $c = (I, Rot(h))(e_1, e_2)^t$ is a syndrome of a double-circulant code.

$\rightarrow$ small key size and ciphertext of size n, 4800 for 80b, 13200b for 128b security ($\sim$ 14.000 with last evulation of DFR)

Security analysis

1. **Generalized McEliece** structural attacks : the system is quasi-cyclic and relies on the masking of a trapdoor decoding, ntru-like matrix, generated by small weight vectors : indistinguashibility of the MDPC matrix : "almost random codes" but not uniquely random, but clearly less structured than other McEliece based systems (no better attack than the search problem).

2. the length the message has same size that the size of the key (or twice depending McEliece or Niederreiter), hence greater than McEliece - Goppa

3. probability of decryption makes it more suitable for key exchange than for pure encryption, long standing problematic of analysis of the very small DFR solved in 2025, weak keys problematic.

4. more generally the decryption problem relies on decoding random double-circulant codes.

Decoding random quasi-cyclic codes

Definition (Quasi-Cyclic Syndrome Decoding problem)

Given a parity check matrix H of QC code C and $y = H.e^t$, with e of weight w. **Question :** is it possible to decode e from y ?

- Same problem than for SD but for a QC matrix.

- In practice : the best known attack cannot use the quasi-cyclicity (gain in n from the DOOM (Decoding One Out of Many, Sendrier '15)

- Search to decision reduction ? ? classical approach does not work because of QC structure (not yet but see Bombar, Couvreur, Debris-Alazard : On Codes and Learning with Errors over Function Fields(2021)).

- Better quantum algorithm ? ? ? Case of ideal-lattice different ? ?

Aleknovich's approach

Alekhnovich first cryptosystem (2003)

Hypothesis : Decisional Decoding Hypothesis with parameter t

Let $0 < R_1 < R_2 < 1$. Consider k, n such that $R_1 \leq k/n \leq R_2$, $\mathcal{C}$ a random code (generated by $\mathbf{G}$), and a vector which is either :

(i) a uniformly random vector $\mathbf{u}$

(ii) $\mathbf{c} + e$ where $\mathbf{c} \in \mathcal{C}$ is a unif. rand. codeword and e a unif. rand. error of weight t, ind. of $\mathbf{c}$.

There is no polynomial-time decoding algorithm $\mathcal{A}$ that decides between (i) and (ii) with a non-negligible advantage over random choice.

Key generation

- $\mathbf{A} \leftarrow \mathbb{F}_2^{k \times n}$, $\mathbf{e} \leftarrow \mathcal{S}_t^n(\mathbb{F}_2)$
- $\mathbf{y} \leftarrow \mathbf{x}\mathbf{A} + \mathbf{e}$, for $\mathbf{x} \leftarrow \mathbb{F}_2^n$
- $\mathtt{pk} = \mathbf{H} \leftarrow \begin{pmatrix} \mathbf{A} \\ \hline \mathbf{y} \end{pmatrix}$
- $\mathtt{sk} = \mathbf{e}$
- $\mathbf{G} = \mathbf{H}^{\perp}$ generating $\mathcal{C}$

Alekhnovich first cryptosystem

Encryption : $c = (c_0$ **or** $c_1)$

- $\mathbf{c}_0 = \mathrm{Enc}_{\mathrm{pk}(0)=\mathbf{u}\leftarrow\mathbb{F}_2^n}$
- $\mathbf{c}_1 = \mathrm{Enc}_{\mathrm{pk}(1)=\mathbf{c}'+\mathbf{t},(\mathbf{c}',\mathbf{t})\leftarrow\mathcal{C}\times\mathcal{S}_t{}^n(\mathbb{F}_2)}$

Decryption

- Return $b = \langle \mathbf{e}, \mathbf{c} \rangle$

$(\langle \mathbf{e}, \mathbf{c}_1 \rangle = \langle \mathbf{e}, \mathbf{c}' \rangle + \langle \mathbf{e}, \mathbf{t} \rangle = \langle \mathbf{e}, \mathbf{t} \rangle)$

Why it works : decryption of c_0 gives randomly 0 or 1 since c_0 is random, when decryption of c_1 gives 0 with strong proba ($\approx 1 - \frac{w(e).w(t)}{n}$ for $w(e)$ and $w(t)$ small). This bias can then be repeated to obtain a correct decryption **but with always a decryption failure**

Security reduction

Alekhnovich shows that a distinguisher between $\mathbf{u}$ and $\mathbf{c} + \mathbf{e}$ yields an algorithm to decode up to t errors.

$\rightarrow$ Can be adapted to larger plaintext spaces, but results in inefficient cryptosystems. Meanwhile it shows that it is possible to rely the security to distinguish between a random element and a codeword plus an error : hence the DecisionalSD problem which is polynomially equivalent to search SD

$\rightarrow$ cryptosystem with a reduction to decoding RANDOM codes

Noisy Diffie-Hellman and Hamming Quasi-Cyclic (HQC) scheme

Noisy Diffie-Hellman : the idea behind HQC)

G.,Aguilar,Zemor et al., Noisy Diffie-Hellman, first presented at the Rump Session PQCrypto 2010 - Darmstadt

• DH : $A \rightarrow g^a \pmod p$, $B \rightarrow g^b \pmod p$
Why does it work ? $(g^a)^b = (g^b)^a = g^{ab}$
Double application of $x \rightarrow g^x$ + commutativity

Analogy : $x \rightarrow g^x$ and $x \rightarrow He^t$

Problem of double application ? ? ? $\rightarrow$ consider H of rate $1/2$
Problem of commutativity ? ? ? $\rightarrow$ Double circulant codes

H=(1,h) (double circulant code) : multiplication mod $x^n - 1$

Alice (x_A, y_A) $\rightarrow s_A = x_A + h.y_A \rightarrow$ **Bob**

Alice $\leftarrow s_B = x_B + h.y_B \leftarrow$ **Bob**(x_B, y_B)

Alice : $y_A.s_B = y_A x_B + y_A h y_B = y_A.s_B = y_A x_B + \mathbf{h y_A y_B}$
Bob :$y_B.s_A = y_B x_A + y_B h y_A = y_B.s_A = y_B x_A + \mathbf{h y_A y_B} \Rightarrow$ Alice and Bob share the same "random" value $\mathbf{h y_A y_B}$ + a different noisy part $y_A x_B$ or $y_B x_A$ which can be made small.

HQC : Aguilar, Blazy, Deneuville, Gaborit, Zemor (2016), based on a previous noisy DH version of 2010

Encryption scheme in **H**amming metric, using **Q**uasi-**C**yclic Codes, a El-Gamal version of the noisy DH protocol.

- Notation : Secret data - Public data - One-time Randomness
- **G** is the generator matrix of some public code $\mathcal{C}$.

Alice		Bob
$\mathsf{seed_h} \leftarrow \{0,1\}^\lambda$, $\mathbf{h} \xleftarrow{\mathsf{seed_h}} \mathbb{F}_2^n$		
$\mathbf{x}, \mathbf{y} \leftarrow \mathcal{S}_w^n(\mathbb{F}_2)$, $\mathbf{s} \leftarrow \mathbf{x} + \mathbf{hy}$	$\xrightarrow{\mathsf{seed_h}, \mathbf{s}}$	$\mathbf{r}_1, \mathbf{r}_2 \leftarrow \mathcal{S}_w^n(\mathbb{F}_2)$, $\epsilon \leftarrow \mathcal{S}_{cw}^n(\mathbb{F}_2)$
		$\mathbf{v} \leftarrow \mathbf{r}_1 + \mathbf{hr}_2$,
	$\xleftarrow{\mathbf{v}, \rho}$	$\rho \leftarrow \mu\mathbf{G} + \mathbf{sr}_2 + \epsilon$
$\mu \leftarrow \mathcal{C}.\mathsf{Decode}(\rho - \mathbf{vy})$		

Correctness

Correctness Property

$$\text{Decrypt}(\mathsf{sk}, \text{Encrypt}(\mathsf{pk}, \boldsymbol{\mu}, \theta)) = \boldsymbol{\mu}$$

$$\boldsymbol{\rho} - \mathbf{v} \cdot \mathbf{y} = \mu G + \mathbf{s} \cdot \mathbf{r}_2 - \mathbf{v} \cdot \mathbf{y} + \boldsymbol{\epsilon} = \mu G + (x + hy) r_2 + \epsilon - (r_1 + h r_2) y$$

$$= \mu G + x r_2 + \epsilon - r_1 y + (hy r_2 - h r_2 y) = \mu G + x r_2 + \epsilon - r_1 y$$

$\Rightarrow$ Decodes whenever Weight$(x r_2 + \epsilon - r_1 y)$ is small enough.

Error weight of order $2w(x).w(r_1)$ hence take
$w(x) = w(y) = w(r_1) = w(r_2) = O(\sqrt{n})$.

Tight security reduction for IND-CPA to decoding the 3-QC random matrix

$$\begin{pmatrix} \mathsf{I} & 0 & \mathsf{h} \\ 0 & \mathsf{I} & \mathsf{s} \end{pmatrix}$$

Theorem

HQC is IND-CPA under the 2-DQCSD and 3-DQCSD assumptions.

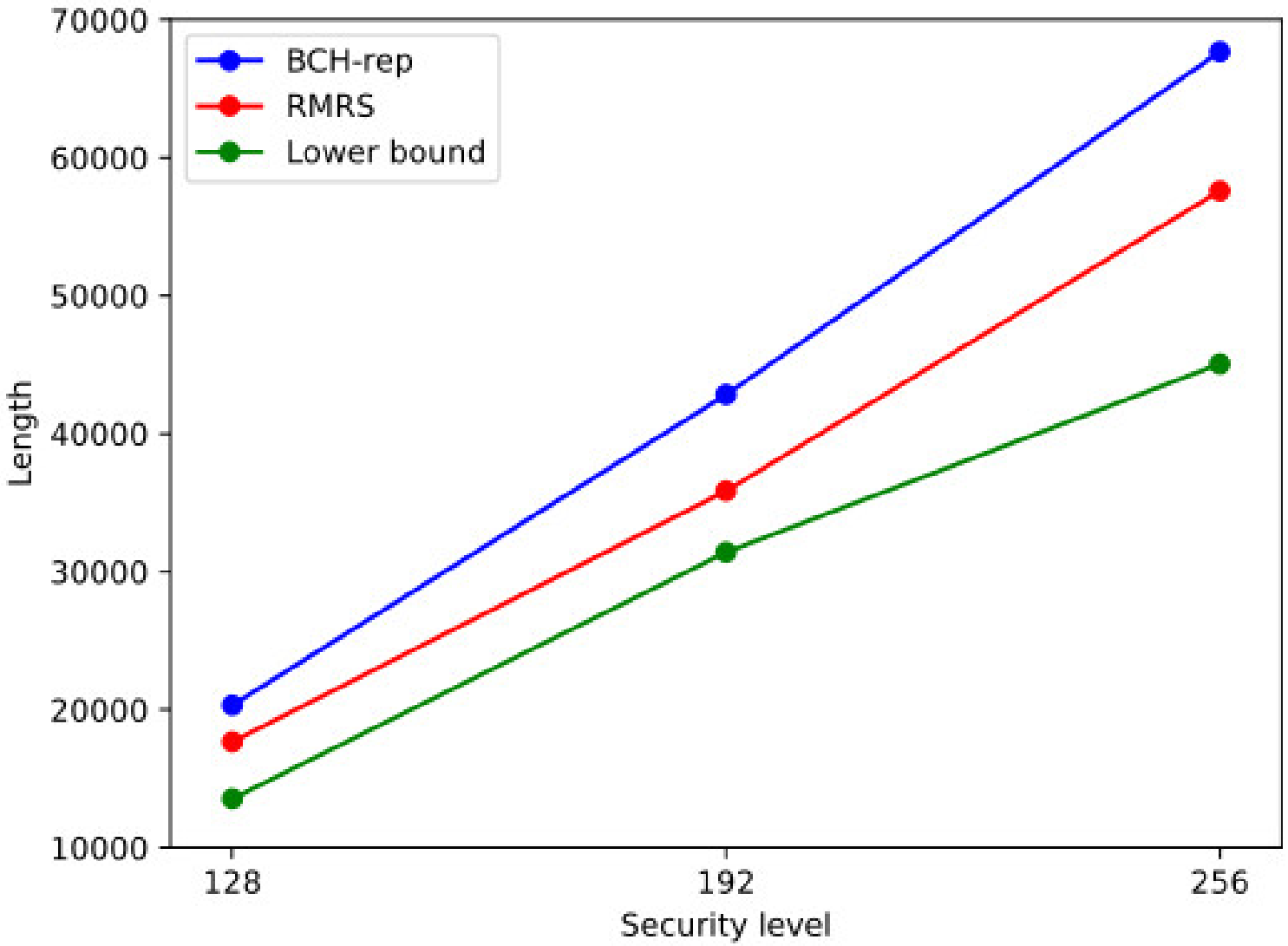
Two codes involved

1. a random double-circulant code on which the security is based
2. a **publicly known** good code G to decode the *large* error

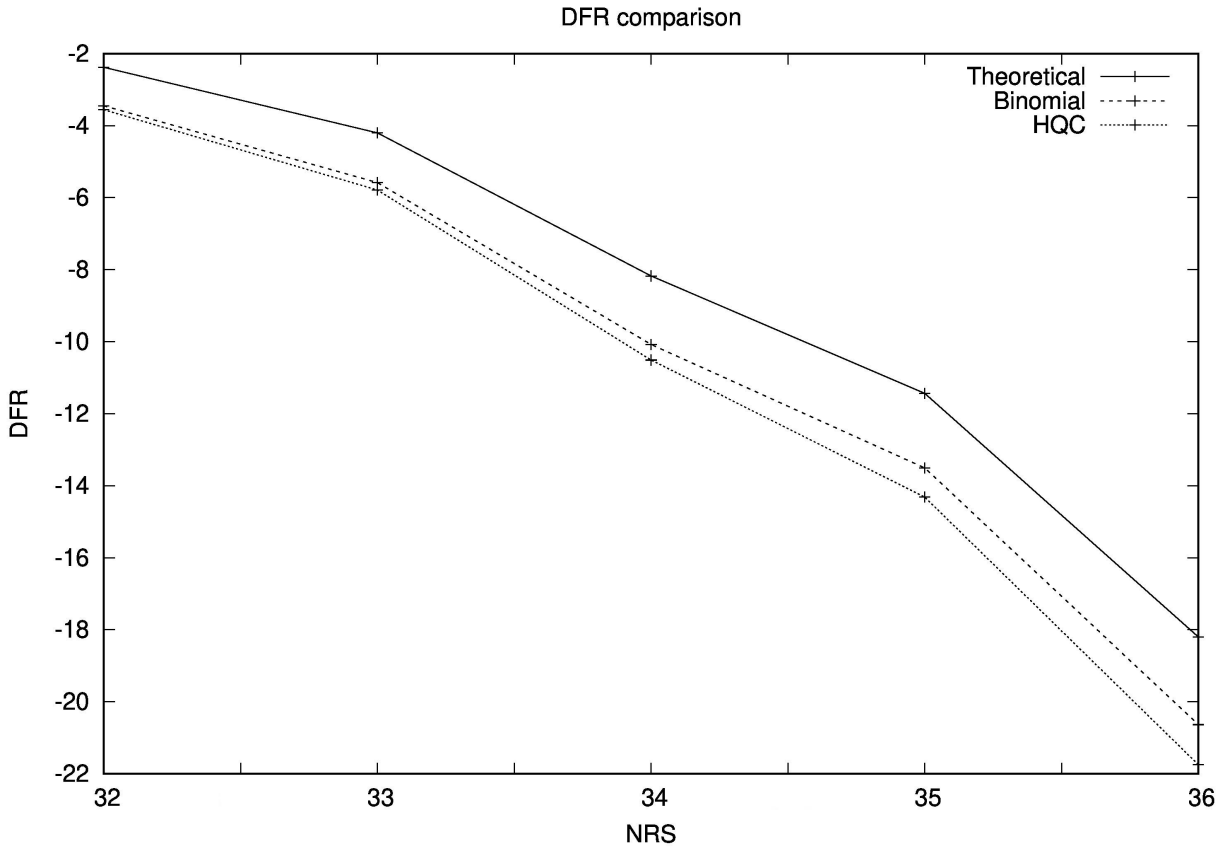
Choice of public good G

1. For the choice of code G : you can choose what you want !
2. if G too simple : easy to analyze but not efficient to decode
3. if G is too complex : very good decoding (hence better parameters) but difficulty to analyze the error probability
4. **Proposed trade-off : (still possible to potentially improve)**
    1. (2016) Tensor code repetition code and (shortened)-BCH codes : efficient + reasonable analysis under reasonable modelization
    2. (2020) Concatenated Reed-Muller and Reed-Solomon codes

Evolution of size for HQC (in bits)



Decryption failure rate



1. Upper bound on the theoretical DFR with a reasonable assumption on the error distribution
2. Observed DFR, with our assumption (practice/theoretical) vs HQC
3. recent improvement of the upper bound : **Antoine Mesnard, Jean-Pierre Tillich, and Valentin Vasseur. The syndrome weight distribution in quasi-cyclic codes, applications to BIKE and HQC (PQCrypto 2026).**

Comparison QC-MDPC/HQC and variants

QC-MDPC/BIKE : key size n, message n (ONE block), security based 2-QCSD + indistingushability of ntru-type matrix, cost of inversion, difficult DFR evaluation, the decoding algo is hidden in the public matrix (no choice), decoding of an error of weight $O(\sqrt{n})$.

HQC : key size n'$\approx 2n$, message 2n' (TWO blocks), security only based on random QC codes, no inversion, good analysis of decryption failure, factor 5 vs BIKE for decoding, error of weight $0(n)$ BUT choice of the decoding algo.

variation between BIKE and HQC : proposal to mix both properties : OUROBOROS, CAKE, CYCLADE, LyuChen : number of blocks, type of security, type of decoding.

The NIST PQ competition for Hamming encryption/key exchange

What did the NIST competition taught us ?

1. Beginning 2017, 2nd Round 2019, 3rd Round 2020 (KYBER), 4th Round 2022 : (McEliece/BIKE/HQC+broken SIKE) $\rightarrow$ **HQC chosen 2025)**
2. the use of KEM : key exchange or small size encryption gives k (128b) then c=m XOR keystream(k) $\rightarrow$ no high encryption rate needed (Niederreiter vs McEliece)
3. Proof in the ROM, permits to reach IND-CCA2 (HHK transform)
4. Lvl 1 : 128b $\rightarrow$ 143 classical bits
5. difference between encryption / key exchange = DFR
6. Constant time implementation
7. Things change (BIKE-2 more efficient inversion / improvement of HQC / systems are attacked / ...)
8. + a lot of other things ...

A quick view on NIST's candidate in Hamming metric

1. McEliece Goppa (PKE)
   - Classic McEliece
   - NTS-KEM (company with patent behind) (merge with Classic MCEliece)
2. QC-McEliece (PKE)
   - BIGQUAKE (2009 QC-McEliece scheme with very conservative parameters and Goppa codes, Gain : factor 8 versus McEliece-Goppa)
   - DAGS (Goppa dyadic with very aggressive parameters (a few kB ... broken, new parameters given and broken again)
3. HQC (PKE)
   - original HQC
   - Lepton : HQC variation (LPN), very small parameters, but badly chosen on LPN - broken
4. M/L-DPC-based (KEM)
   - BIKE (BIKE-1 : CAKE, BIKE-2 : MDPC, BIKE-3 : OUROBOROS), eventually BIKE-2
   - QC-MDPC KEM (original of MDPC paper from a Canadian company, merged refused)
   - LedaKem , variation on original scheme by Baldi et al. , BIKE-2 better on all points.
5. RS variation (PKE)
   - RLCE : half parameters broken, completely inefficient
   - Edon-K : broken via rank metric LRPC decoding
6. Signatures : Ramstake, pqsignRM : broken (or will be) and inefficient

Scheme	*pk*	ct	pk+ct	KeyGen	Encaps	Decaps
KYBER	800 B	768 B	1,568 B	33	45	60
BIKE	1,540 B	1,572 B	3,112 B	589	97	1,135
HQC	2,249 B	4,481 B	6,720 B	75	175	356
Classic McEliece	261 kB	96 B	261 kB	140,870	46	137

Figure – Comparison of different schemes for 128 bits of security, speed in kcycles for optimized implem

Main use : TLS : ct+pk, size may not matter so much, depends of the transmission rate, for TLS : KYBER/HQC/BIKE in term of efficiency for regular use, with regular transmission.

CONCLUSION for (Hamming) Code-based encryption

1. Lot of things done since McEliece '78, especially recently
2. Satisfying, fast with small parameters : BIKE and HQC
3. Big theoretical open questions : advanced encryption : efficient homomorphic schemes, IBE, etc...
4. Implementation work : side-channel attacks, secure implemention (software/hardware)
5. New recent question on the security of McEliece

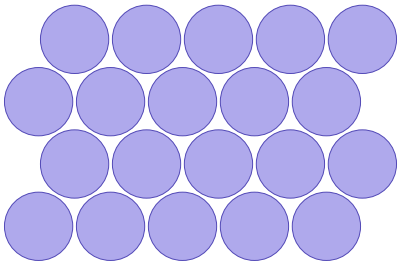
Authentication and Signature for Hamming code-based crypto

Different approaches for signature

1. Signatures by inversion (Full Domain Hash)
    1. unique pre-image : RSA,CFS
    2. several pre-images : NTRUSign, GGH, GPV, WAVE, MIRANDA
2. Signature by proof of knowledge
    1. by construction (pre-image of a structured syndrome) : Schnorr, DSA, Lyubashevski (lattices 2012) (Di-Lithium)
    2. generic : ZK authentication + Fiat-Shamir paradigm : SDITH, LESS, RYDE, MIRATH
3. one-time signatures : KKS '97, Lyubashevski '07

CFS signature scheme 2001

1. In 2001 Courtois Finiasz and Sendrier proposed a RSA-like signature for codes.
2. SUppose one gets a decoding algorithm (inverse of a syndrome) unlike RSA, is it possible to consider a random codeword and invert it ? ?
3. in general no : the density of invertible codewords is exponentially low.... for a Goppa $[2^m, 2^m - tm, 2t + 1]$ the density is $\frac{1}{t!}$
4. CFS '01 : consider extreme parameters where t is low (≤ 10) and repeat until it works !
5. leads to 'flat' hidden matrices : $[2^{16}, 154]$, signature shorts, very large size of keys : several MB, rather lengthy - more than RSA -(but can be parallelized),
6. scales badly (exponential size of public key in term of security parameter), distinguiashable, recently parameters attacked (Bardet et al. 2026)



WAVE signature scheme 2019

Debris-Sendrier-Tillich (WAVE 2019)

In the spirit of GPV for lattices :

- proof of no leakeage
- based on Large Weight in GF(3)
- rather large public ket (2MB) but rather small signature (700B)

Fiat-Shamir paradigm

1. Starts from a ZK authentication scheme (cheating proba p) and turns into a signature scheme
2. Idea : the prover replaces the verifier by a hash function.
    1. Fix a security level and t rounds
    2. Prepare t commitments at once $\rightarrow C$
    3. consider a sequence of challenges $b(b_1, ..., b_t)$ as $b \leftarrow hash(C)$
    4. compute the sequence of answers $A = (A_1, A_2, ..., A_t)$
    5. the signature is : (C, A).
    6. verification : the verifier checks that A is the lists of answers regarding C

$\Rightarrow$ usually p is $1/2$ or $2/3$ hence the signature is long .... but it works, small public keys

Very short keys for the Stern authentication scheme

Stern authentication protocol

Zero-knowledge protocol with cheating probability 2/3, Crypto '93
H a $(n-k) \times n$ binary matrix
Secret key : H and x a word of length n and weight w such that $w < d_{VG}(n)$.
Public key : H and s (syndrom) : $s = Hx^T$

Protocol :

1. P (prover) chooses randomly y of length n and a permutation σ on $\{1, \cdots, n\}$. Send to V (verifier) : c_1, c_2 and c_3 such that : $c_1 =< \sigma, Hy^T >; c_2 =< \sigma(y) >; c_3 =< \sigma(x+y) >$ for $< arg_1, arg_2 >$ the action of a hash function on the concatenation of arg_1 and arg_2, and $arg.\sigma$ the image of arg with σ.

2. V sends to P a challenge b in $\{0, 1, 2\}$.

3. P receives b, then Three possibilities occur :

   - if $b = 0$: P reveals y and σ,
   - if $b = 1$: P reveals $(y + x)$ and σ,
   - if $b = 2$: P reveals $\sigma(y)$ and $\sigma(x)$.

4. Three possibilities occur :

   - if $b = 0$: V checks that c_1 and c_2 received are correct.
   - if $b = 1$: V checks that c_1 and c_3 receveid are correct.
   ONe can remark that : $Hy^T = H(y + x)^T + s$
   - if $b = 2$: V checks that c_2 and c_3 received are correct, and $weight(\sigma(x)) = w$.

5. Repeat steps $1, 2, 3$ and 4 until the adequate security level is reached.

Zero-knowledge protocol with cheating probability at most 2/3

Security : Random matrix :

Theoretical

- Satisfies Gilbert-Varshamov bound (existence)
- Almost all random codes are on GV (parameters)
- NP-complete problem

Advantage : no masking !

Improvement Aguilar Gaborit Schrek '10 : 2/3 $\rightarrow$ 1/2 $\rightarrow$ signature size $\approx$ 20kB

Improvement for knowledge proofs

Classical ZK approach : use properties of the problem for obtaining ZK : x+random, permutation(x) $\rightarrow$ hard to decrease the cheating proba below 1/2

New (old) approach : MPC in the head (2007) + new results (PICNIC) : permits to decrease the cheating proba in 1/N at the cost of adding symmetric crypto through Merkkle trees.

Idea : prove the knowledge of solution of multivariable equations (typically degree 2).

New game : characterize Hx=s in term of equations of lowest degree (Modeling) + apply MPC in the head approach

First approach : Feneuil, Joux, Rivain : Syndrome Decoding in the Head : Shorter Signatures from Zero-Knowledge Proofs(2022) (9kB) (SDITH NIST)

- **Improvement on MPC in the head (threshold) :** Thibauld Feneuil, Matthieu Rivain : Threshold Computation in the Head : Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments (2023) (similar to Vole in the head) (6kB)

Improvement on modeling : Slim Bettaieb, Loic Bidoux, Philippe Gaborit, Mukul Kulkarni : Modelings for generic PoK and Applications : Shorter SD and PKP based Signatures (3.8kB) (SDITH -2)

Advantage of MPC in the head approach :

- proba cheating in 1/N, in Hamming : signature size : 3.8kB + pk : 100B
- fast : ($\approx$ 1ms)
- relies on purely generic problem with the most difficult instances (GV)
- relatively resilient, increasing security parameter has a small impact on the size

Limitations

- quadratic size of signature x2 secu implies x4 on signature size

Difference Knowledge proof/ Full Domain Hash

- larger signature sizes (3-4 kB), small pk (100B), fast, generic problem security / very small signature (100B), (large ?) pk 500kB, fast (?), trapdoor security (structural attacks)

Another approach not based on SD : LESS

Alessandro Barenghi, Jean-Francois Biasse, Edoardo Persichetti, Paolo Santini : LESS-FM : Fine-Tuning Signatures from the Code Equivalence Problem. PQCrypto 2021

Code equivalence problem : C a code and a permutation P, recover P from CP.

1. ZK based on Code-equivalence problem rather than SD (Stern)
2. In that case possibility to decrease below 1/2 in fact (1/N) but at the price of increasing the size of the public key (second round of NIST signature but not third)
3. based on code equivalence on codes over q=127

Conclusion on Hamming code-based cryptography

- In a few years : big progress both in term of encryption and signature
- Efficient Full domain hash signature and/or very small size signature ?
- MPC opens the door to many variations on signature : ring, group, blind, anonymous, etc...Other metrics : rank metric (can do better than lattices in term of size), minrank
- Efficient side-channel counter-measures and attacks.
- Main Open questions : advanced cryptography (homomorphic encryption O(n) messages, IBE, FHE ....)

Questions ? ? ?