

# Code-based Cryptography

---

PQCRYPTO Summer School on Post-Quantum Cryptography 2017

TU Eindhoven

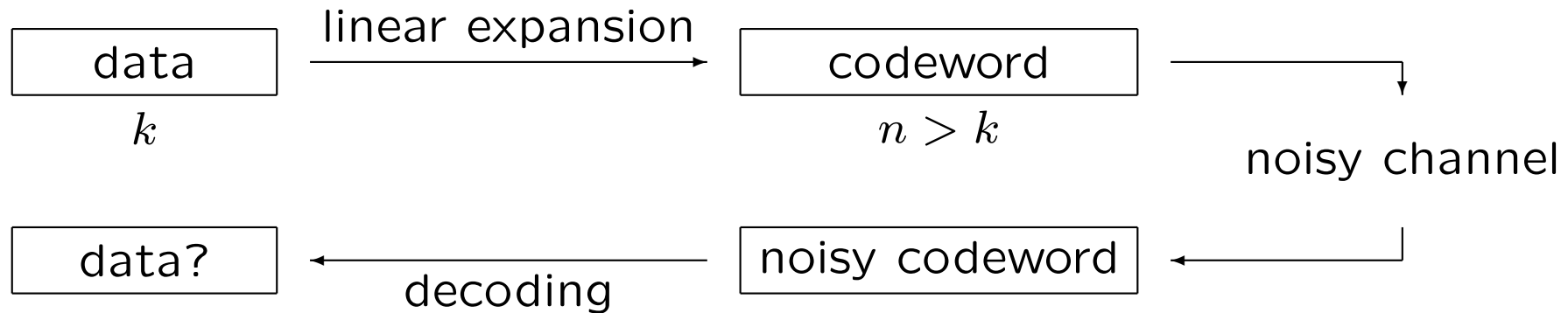
---

Nicolas Sendrier



PQCRYPTO  
ICT-645622

## Linear Codes for Telecommunication



[Shannon, 1948] (for a binary symmetric channel of error rate  $p$ ):  
Decoding probability  $\rightarrow 1$  if  $\frac{k}{n} = R < 1 - h(p)$

( $h(p) = -p \log_2 p - (1 - p) \log_2(1 - p)$  the binary entropy function)

Codes of rate  $R$  can correct up to  $\lambda n$  errors ( $\lambda = h^{-1}(1 - R)$ )

For instance 11% of errors for  $R = 0.5$

**Non constructive**  $\rightarrow$  no poly-time algorithm for decoding in general

# Random Codes Are Hard to Decode

When the linear expansion is random:

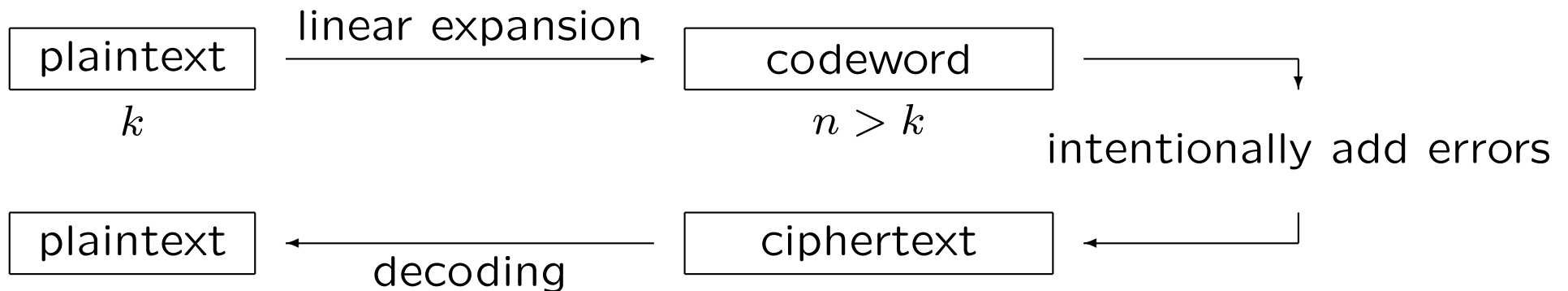
- Decoding is NP-complete [Berlekamp, McEliece & van Tilborg, 78]
- Even the tiniest amount of error is (believed to be) hard to remove. Decoding  $n^\varepsilon$  errors is conjectured difficult on average for any  $\varepsilon > 0$  [Alekhovich, 2003].

## Codes with Good Decoders Exist

Coding theory is about finding “good” codes (i.e. linear expansions)

- alternant codes have a poly-time decoder for  $\Theta\left(\frac{n}{\log n}\right)$  errors
- some classes of codes have a poly-time decoder for  $\Theta(n)$  errors (algebraic geometry, expander graphs, concatenation, ...)

## Linear Codes for Cryptography



- If a random linear code is used, no one can decode efficiently
- If a “good” code is used, anyone who knows the structure has access to a fast decoder

Assuming that the knowledge of the linear expansion does not reveal the code structure:

- The linear expansion is public and anyone can encrypt
- The decoder is known to the legitimate user who can decrypt
- For anyone else, the code looks random

# Why Consider Code-Based Cryptography?

Because

- it's always good to understand more things
- cryptography needs diversity to evolve against
  - quantum computing
  - algorithmic progress
- we can do it
  - that's what those lectures are about

# Outline

- I. Introduction to Codes and Code-based Cryptography
- II. Instantiating McEliece
- III. Security Reduction to Difficult Problems
- IV. Implementation
- V. Practical Security - The Attacks
- VI. Other Public Key Systems

# I. Introduction to Codes and Code-based Cryptography



## Notations

$\mathbf{F}_q$  the finite field with  $q$  elements

Hamming distance:  $x = (x_1, \dots, x_n) \in \mathbf{F}_q^n$ ,  $y = (y_1, \dots, y_n) \in \mathbf{F}_q^n$

$$\text{dist}(x, y) = |\{i \in \{1, \dots, n\} \mid x_i \neq y_i\}|$$

Hamming weight:  $x = (x_1, \dots, x_n) \in \mathbf{F}_q^n$ ,

$$|x| = |\{i \in \{1, \dots, n\} \mid x_i \neq 0\}| = \text{dist}(x, \mathbf{0})$$

$$\mathcal{S}_n(\mathbf{0}, t) = \{e \in \mathbf{F}_q^n \mid |e| = t\}$$

(the sphere, in the Hamming space  $\mathbf{F}_q^n$ , centered in  $\mathbf{0}$  of radius  $t$ )

# Linear Error Correcting Codes

A  $q$ -ary linear  $[n, k]$  code  $\mathcal{C}$  is a  $k$ -dimensional subspace of  $\mathbf{F}_q^n$

A generator matrix  $G \in \mathbf{F}_q^{k \times n}$  of  $\mathcal{C}$  is such that  $\mathcal{C} = \{xG \mid x \in \mathbf{F}_q^k\}$

It defines an encoder for  $\mathcal{C}$

$$\begin{aligned} f_G : \mathbf{F}_q^k &\rightarrow \mathcal{C} \\ x &\mapsto xG \end{aligned}$$

The encoding can be inverted by multiplying a word of  $\mathcal{C}$  by a right inverse  $G^*$  of  $G$ : if  $GG^* = \text{Id}$  then  $f_G(x)G^* = xGG^* = x$

If  $G$  is in systematic form,  $G = (\text{Id} \mid R)$  then  $G^* = (\text{Id} \mid \mathbf{0})^T$  is a right inverse and the de-encoding consists in truncating

## Parity Check Matrix and Syndrome

Let  $\mathcal{C}$  be a  $q$ -ary linear  $[n, k]$  code, let  $r = n - k$

A parity check matrix  $H \in \mathbf{F}_q^{r \times n}$  of  $\mathcal{C}$  is such that  $\mathcal{C} = \{x \in \mathbf{F}_q^n \mid xH^T = 0\}$

The  $H$ -syndrome (or syndrome) of  $y \in \mathbf{F}_q^n$  is  $S_H(y) = yH^T$

For all  $y \in \mathbf{F}_q^n$ , let  $s = yH^T$ , the coset of  $y$  is defined as

$$\text{Coset}(y) = y + \mathcal{C} = \{z \in \mathbf{F}_q^n \mid zH^T = yH^T = s\} = S_H^{-1}(s)$$

The cosets form a partition of the space  $\mathbf{F}_q^n$

## Decoding and Syndrome Decoding

Let  $\mathcal{C}$  be a  $q$ -ary linear  $[n, k]$  code, let  $H$  be a parity check matrix of  $\mathcal{C}$

- $\Phi_{\mathcal{C}} : \mathbb{F}_q^n \rightarrow \mathcal{C}$  is a  $t$ -bounded decoder if for all  $x \in \mathcal{C}$  and all  $e \in \mathbb{F}_q^n$

$$|e| \leq t \Rightarrow \Phi_{\mathcal{C}}(x + e) = x$$

- $\Psi_H : \mathbb{F}_q^{n-k} \rightarrow \mathbb{F}_q^n$  is a  $t$ -bounded  $H$ -syndrome decoder if for all  $e \in \mathbb{F}_q^n$

$$|e| \leq t \Rightarrow \Psi_H(eH^T) = e$$

$\exists$  an efficient  $t$ -bounded decoder  $\Leftrightarrow \exists$  an efficient  $t$ -bounded syndrome decoder

## McEliece Public-key Encryption Scheme – Overview

Let  $\mathcal{F}$  be a family of  $t$ -error correcting  $q$ -ary linear  $[n, k]$  codes

### Key generation:

pick  $\mathcal{C} \in \mathcal{F} \rightarrow \left\{ \begin{array}{l} \text{Public Key: } G \in \mathbf{F}_q^{k \times n}, \text{ a generator matrix} \\ \text{Secret Key: } \Phi : \mathbf{F}_q^n \rightarrow \mathcal{C}, \text{ a } t\text{-bounded decoder} \end{array} \right.$

**Encryption:**  $\left[ \begin{array}{l} E_G : \mathbf{F}_q^k \rightarrow \mathbf{F}_q^n \\ x \mapsto xG + e \end{array} \right]$  with  $e$  random of weight  $t$

**Decryption:**  $\left[ \begin{array}{l} D_\Phi : \mathbf{F}_q^n \rightarrow \mathbf{F}_q^k \\ y \mapsto \Phi(y)G^* \end{array} \right]$  where  $GG^* = 1$

*Proof:*  $D_\Phi(E_G(x)) = D_\Phi(xG + e) = \Phi(xG + e)G^* = xGG^* = x$

## Niederreiter Public-key Encryption Scheme – Overview

Let  $\mathcal{F}$  be a family of  $t$ -error correcting  $q$ -ary  $[n, k]$  codes,  $r = n - k$

**Key generation:** pick  $\mathcal{C} \in \mathcal{F}$

→  $\left\{ \begin{array}{l} \text{Public Key: } H \in \mathbf{F}_q^{r \times n}, \text{ a parity check matrix} \\ \text{Secret Key: } \Psi : \mathbf{F}_q^r \rightarrow \mathbf{F}_q^n, \text{ a } t\text{-bounded } H\text{-syndrome decoder} \end{array} \right.$

**Encryption:**  $\left[ \begin{array}{l} E_H : \mathcal{S}_n(\mathbf{0}, t) \rightarrow \mathbf{F}_q^r \\ e \mapsto eH^T \end{array} \right]$

**Decryption:**  $\left[ \begin{array}{l} D_\Psi : \mathbf{F}_q^r \rightarrow \mathcal{S}_n(\mathbf{0}, t) \\ s \mapsto \Psi(s) \end{array} \right]$

*Proof:*  $D_\Psi(E_H(e)) = D_\Psi(eH^T) = e$

## McEliece/Niederreiter Security

The following two problems must be difficult enough:

1. Retrieve an efficient  $t$ -bounded decoder from the public key (*i.e.* a generator matrix or a parity check matrix)

The legitimate user must be able to decode thus some structure exists, it must remain hidden to the adversary

2. Decode  $t$  errors in a random  $q$ -ary  $[n, k]$  code

Without knowledge of the trapdoor the adversary is reduced to use generic decoding techniques

The parameters  $n$ ,  $k$  and  $t$  must be chosen large enough

## In Practice

[McEliece, 1978]

“A public-key cryptosystem based on algebraic coding theory”

The secret code family consisted of irreducible binary Goppa codes of length 1024, dimension 524, and correcting up to 50 errors

- public key size: 536 576 bits
- cleartext size: 524 bits
- ciphertext size: 1024 bits

A bit undersized today (attacked in [Bernstein, Lange, & Peters, 08] with  $\approx 2^{60}$  CPU cycles)

[Niederreiter, 1986]

“Knapsack-type cryptosystems and algebraic coding theory”

Several families of secret codes were proposed, among them Reed-Solomon codes, concatenated codes and Goppa codes. Only Goppa codes are secure today.



## II. Instantiating McEliece

## Which Code Family ?

Finding families of codes whose structure cannot be recognized seems to be a difficult task

Family	Proposed by	Broken by
Goppa	McEliece (78)	-
Reed-Solomon	Niederreiter (86)	Sidelnikov & Chestakov (92)
Concatenated	Niederreiter (86)	Sendrier (98)
Reed-Muller	Sidelnikov (94)	Minder & Shokrollahi (07)
AG codes	Janwa & Moreno (96)	Faure & Minder (08) Couvreur, Márquez-Corbella. & Pellikaan (14)
LDPC	Monico, Rosenthal, & Shokrollahi (00)	
Convolutional codes	Löndahl & Johansson (12)	Landais & Tillich (13)

[Faugère, Gauthier, Otmani, Perret, & Tillich, 11] distinguisher for binary Goppa codes of rate  $\rightarrow 1$

## More on Goppa Codes

Goppa codes are not limited to the binary case. It is possible to define  $q$ -ary Goppa codes with a support in  $\mathbf{F}_q^m$ .

[Bernstein, Lange, & Peters, 10]: Wild McEliece. The key size can be reduced in some case. There are limits:

- [Couvreur, Otmani, & Tillich, 14] Choose  $m > 2$
- [Faugère, Perret, & Portzamparc, 14] Caution if  $q$  not prime

## Reducing the Public Key Size

In a block-circulant matrix, each (square) block is completely defined by its first row  $\rightarrow$  public key size is linear instead of quadratic

$$G = \begin{array}{|c|c|c|} \hline \boxed{g_{0,0}} & \boxed{g_{0,1}} & \boxed{g_{0,2}} \\ \hline \circlearrowright & \circlearrowright & \circlearrowright \\ \hline \boxed{g_{1,0}} & \boxed{g_{1,1}} & \boxed{g_{1,2}} \\ \hline \circlearrowright & \circlearrowright & \circlearrowright \\ \hline \end{array}$$

- Quasi-cyclic [Gaborit, 05] or quasi-dyadic [Misoczki & Barreto, 09] alternant (Goppa) codes. Structure + structure must be used with great care [Faugère, Otmani, Perret, & Tillich, 10]
- Disguised QC-LDPC codes [Baldi & Chiaraluce, 07]. New promising trend.
- QC-MDPC [Misoczki, Tillich, Sendrier, & Barreto, 13]. As above with a stronger security reduction.

## Irreducible Binary Goppa Codes

Parameters:  $m, t$  and  $n \leq 2^m$

Support:  $L = (\alpha_1, \dots, \alpha_n)$  distinct in  $\mathbb{F}_{2^m}$

Generator:  $g(z) \in \mathbb{F}_{2^m}[z]$  monic irreducible of degree  $t$

For all  $a = (a_{\alpha_1}, \dots, a_{\alpha_n}) \in \mathbb{F}_2^n$  (we use  $L$  to index the coordinates) let

$$R_a(z) = \sum_{\beta \in L} \frac{a_\beta}{z - \beta} \text{ and } \sigma_a(z) = \prod_{\beta \in L} (z - \beta)^{a_\beta}.$$

The binary irreducible Goppa code  $\Gamma(L, g)$  is defined by

$$a \in \Gamma(L, g) \Leftrightarrow R_a(z) = 0 \pmod{g(z)}.$$

It is a binary linear  $[n, k \geq n - mt]$  code and for all  $e \in \mathbb{F}_2^n$

$$R_e(z)\sigma_e(z) = \frac{d}{dz}\sigma_e(z) \pmod{g(z)}. \quad (1)$$

Given  $R_e(z)$ , the key equation (1) can be solved in  $\sigma_e(z)$  if  $|e| \leq t$  providing a poly-time  $t$ -bounded decoder.

## Some Sets of Parameters for Goppa Codes

$m, t$	text size in bits				key size	message security*
	McEliece		Niederreiter			
	cipher	clear	cipher	clear		
10, 50	1024	524	500	284	32 kB	52
11, 40	2048	1608	440	280	88 kB	81
12, 50	4096	3496	600	385	277 kB	120

\* logarithm in base 2 of the cost of the best known attack  
lower bound derived from ISD, BJMM variant (generic decoder)

the key security is always higher ( $\approx mt$ )

key size is given for a key in systematic form

## Some Sets of Parameters for QC-MDPC-McEliece

Binary QC-MDPC  $[n, k]$  code with parity check equations of weight  $w$  correcting  $t$  errors

$(n, k, w, t)$	size in bits			security*	
	cipher	clear	key	message	key
(9602, 4801, 90, 84)	9602	4801	4801	80	79
(19714, 9857, 142, 134)	19714	9857	9857	128	129

\* logarithm in base 2 of the cost of the best known attack  
lower bound derived from ISD, BJMM variant

The best key attack and the best message attack are both based on generic decoding

# III. Security Reduction to Difficult Problems



## Hard Decoding Problems

[Berlekamp, McEliece, & van Tilborg, 78]

### Syndrome Decoding

NP-complete

*Instance:*  $H \in \mathbb{F}_2^{r \times n}$ ,  $s \in \mathbb{F}_2^r$ ,  $w$  integer

*Question:* Is there  $e \in \mathbb{F}_2^n$  such that  $|e| \leq w$  and  $eH^T = s$ ?

### Computational Syndrome Decoding

NP-hard

*Instance:*  $H \in \mathbb{F}_2^{r \times n}$ ,  $s \in \mathbb{F}_2^r$ ,  $w$  integer

*Output:*  $e \in \mathbb{F}_2^n$  such that  $|e| \leq w$  and  $eH^T = s$

[Finiasz, 04]

### Goppa Bounded Decoding

NP-hard

*Instance:*  $H \in \mathbb{F}_2^{r \times n}$ ,  $s \in \mathbb{F}_2^r$

*Output:*  $e \in \mathbb{F}_2^n$  such that  $|e| \leq \frac{r}{\log_2 n}$  and  $eH^T = s$

**Open problem:** average case complexity (Conjectured difficult)

## Hard Structural Problems

### Goppa code Distinguishing

NP

*Instance:*  $G \in \mathbb{F}_2^{k \times n}$

*Question:* Does  $G$  span a binary Goppa code?

- NP: the property is easy to check given  $(L, g)$
- Completeness status is unknown
- Easy when the information rate  $\rightarrow 1$   
(Faugère, Gauthier, Otmani, Perret, & Tillich, 11)

### Goppa code Reconstruction

*Instance:*  $G \in \mathbb{F}_2^{k \times n}$

*Output:*  $(L, g)$  such that  $\Gamma(L, g) = \{xG \mid x \in \mathbb{F}_q^k\}$

- Tightness: gap between decisional and computational problems

## Decoders and Distinguishers

For given parameters  $n$ ,  $k$ , and  $t$

Let  $\mathcal{G} \subset \mathcal{K} \subset \mathbb{F}_2^{k \times n}$ , where  $\mathcal{G}$  is the public key space and  $\mathcal{K}$  the apparent public key space. (in the original scheme,  $\mathcal{G}$  is the set of all generator matrices of a Goppa code and  $\mathcal{K} = \mathbb{F}_2^{k \times n}$ )

For quasi-cyclic variants, the apparent key space  $\mathcal{K}$  is limited to block-circulant matrices.

We consider two programs

- a decoding algorithm:  $\mathcal{A} : \mathbb{F}_2^n \times \mathbb{F}_2^{k \times n} \rightarrow \mathcal{S}_n(\mathbf{0}, t)$
- a distinguisher:  $\mathcal{D} : \mathbb{F}_2^{k \times n} \rightarrow \{\text{true}, \text{false}\}$

We consider the sample space  $\Omega = \mathbb{F}_2^k \times \mathbb{F}_2^{k \times n} \times \mathcal{S}_n(\mathbf{0}, t)$  equipped with the uniform distribution, and the event (successful decoding)

$$\mathcal{S}_{\mathcal{A}} = \{(x, G, e) \in \Omega \mid \mathcal{A}(xG + e, G) = e\}$$

## Decoders and Distinguishers (continued)

$\mathcal{K}$  the *apparent* public key space     $\mathcal{A} : \mathbb{F}_2^n \times \mathbb{F}_2^{k \times n} \rightarrow \mathcal{S}_n(\mathbf{0}, t)$   
 $\mathcal{G}$  the (real) public key space     $\mathcal{D} : \mathbb{F}_2^{k \times n} \rightarrow \{\text{true}, \text{false}\}$

---

$\mathcal{A}$  is a  $(T, \varepsilon)$ -decoder (generic for  $\mathcal{K}$ ) if

- running time:  $|\mathcal{A}| \leq T$
- success probability:  $\text{Succ}_{\text{Dec}}(\mathcal{A}) = \Pr_{\Omega}(\mathcal{S}_{\mathcal{A}} \mid G \in \mathcal{K}) \geq \varepsilon$

$\mathcal{A}$  is a  $(T, \varepsilon)$ -adversary (against McEliece) if

- running time:  $|\mathcal{A}| \leq T$
- success probability:  $\text{Succ}_{\text{McE}}(\mathcal{A}) = \Pr_{\Omega}(\mathcal{S}_{\mathcal{A}} \mid G \in \mathcal{G}) \geq \varepsilon$

$\mathcal{D}$  is a  $(T, \varepsilon)$ -distinguisher (for  $\mathcal{G}$  against  $\mathcal{K}$ ) if

- running time:  $|\mathcal{D}| \leq T$
- advantage:  
$$\text{Adv}(\mathcal{D}) = \left| \Pr_{\Omega}(\mathcal{D}(G) \mid G \in \mathcal{K}) - \Pr_{\Omega}(\mathcal{D}(G) \mid G \in \mathcal{G}) \right| \geq \varepsilon$$

# Security Reduction for McEliece

## Theorem

If there exists a  $(T, \varepsilon)$ -adversary then there exists either

- a  $(T, \varepsilon/2)$ -decoder (for  $\mathcal{K}$ ),
- or a  $(T + O(n^2), \varepsilon/2)$ -distinguisher (for  $\mathcal{G}$  against  $\mathcal{K}$ ),

*Proof (hint):*

$\mathcal{D}(G)$ :

$x \leftarrow \mathbf{F}_2^k$  ;  $e \leftarrow \mathcal{S}_n(\mathbf{0}, t)$  // randomly and uniformly  
return  $\mathcal{A}(xG + e, G) \stackrel{?}{=} e$

The result holds also for the Niederreiter scheme and for any real and apparent public key spaces  $\mathcal{G}$  and  $\mathcal{K}$ . For quasi-cyclic variants, the apparent key space  $\mathcal{K}$  is limited to block-circulant matrices.

# One Way Encryption Schemes

A scheme is OWE (One Way Encryption) if all the attacks are intractable on average when the messages and the keys are uniformly distributed

Loosely speaking, there is no  $(T, \varepsilon)$ -adversary with  $T/\varepsilon$  upper bounded by a polynomial in the system parameters

Assuming

- decoding in a random linear code is hard
- Goppa codes are pseudorandom

McEliece and Niederreiter cryptosystems are One Way Encryption (OWE) schemes

## Malleability Attacks

### Create New Ciphertext. folklore

If  $y$  is a ciphertext and  $a$  is a codeword then  $y + a$  is a ciphertext

Not a desirable feature *a priori*...

### Resend-message Attack. [Berson, 97]

The same message  $x$  is sent twice with the same public key  $G$

→ the message can be recovered

### Reaction Attack. [Kobara & Imai, 00] ??

We assume the decryption system can be used as an oracle and behaves differently when

- its input is at distance  $> t$  from the code,
- its input is at distance  $\leq t$  from the code.

→ the oracle can be transformed into a decoder

## Semantically Secure Conversions

Being OWE is a very weak notion of security. In the case of code-based systems, it does not encompass attacks such that the “resend-message attack”, the “reaction attack” or, more generally, attacks related to malleability.

Fortunately, using the proper semantically secure conversion any deterministic OWE scheme can become IND-CCA, the strongest security notion.

McEliece is not deterministic but IND-CCA conversion are possible nevertheless, see [Kobara & Imai, 01] for the first one.

An IND-CPA conversion without random oracle also exists [Nojima, Imai, Kobara & Morozov, 08].



# IV. Implementation

## A Remark on Niederreiter Encryption Scheme

In Niederreiter's system the encryption procedure is:

$$\begin{aligned} E_H : \mathcal{S}_n(\mathbf{0}, t) &\rightarrow \mathbf{F}_2^r \\ e &\mapsto eH^T \end{aligned}$$

The set  $\mathcal{S}_n(\mathbf{0}, t)$  is not very convenient to manipulate data, we would rather have an injective mapping

$$\varphi : \mathbf{F}_2^\ell \rightarrow \mathcal{S}_n(\mathbf{0}, t)$$

with  $\ell < \log_2 \binom{n}{t}$  but as close as possible. In addition, we need  $\varphi$  and  $\varphi^{-1}$  to have a fast implementation.

In that case the encryption becomes  $E_H \circ \varphi$  and the decryption  $\varphi^{-1} \circ \mathcal{D}_\Psi$

Note that  $\varphi$  is also required for the semantically secure conversions of McEliece as we must “mix” the error with the message

# Constant Weight Words Encoding - Combinatorial Solution

[Schalkwijk, 72]

We represent a word of  $\mathcal{S}_n(\mathbf{0}, t)$  by the indexes of its non-zero coordinates  $0 \leq i_1 < i_2 < \dots < i_t < n$  and we define the one-to-one mapping

$$\begin{aligned} \theta : \mathcal{S}_n(\mathbf{0}, t) &\longrightarrow \left[ 0, \binom{n}{t} \right[ \\ (i_1, \dots, i_t) &\longmapsto \binom{i_1}{1} + \binom{i_2}{2} + \dots + \binom{i_t}{t} \end{aligned}$$

This mapping can be inverted by using the formula [Sendrier 02]

$$i \approx (xt!)^{1/t} + \frac{t-1}{2} \text{ where } x = \binom{i}{t}$$

We can encode  $\ell = \lfloor \log_2 \binom{n}{t} \rfloor$  bits in one word of  $\mathcal{S}_n(\mathbf{0}, t)$

The cost is quadratic in  $\ell$

## Constant Weight Words Encoding - Source Coding Solutions

Another approach is to use source coding. We try to find an approximative models for constant weight words which are simpler to encode.

It is possible to design fast (linear time) methods with a minimal loss (one or very few bits per block)

- fastest → variable length encoding
- fast → constant length encoding (implemented in HyMES)

Still not negligible compared to the encryption cost

Regular word (used in code-based hash function FSB) is an extreme example with a very high speed but a big information loss (the model for generating constant weight words is very crude)

## Deterministic Version of McEliece

Hybrid McEliece encryption scheme (HyMES) [Biswas & Sendrier, 08]

Parameters:  $m, t, n = 2^m, \varphi : \mathbf{F}_2^\ell \rightarrow \mathcal{S}_n(\mathbf{0}, t)$

**Secret key:**  $(L, g) \in \mathbf{F}_{2^m}^n \times \mathbf{F}_{2^m}[z]$

where  $\begin{cases} L = (\alpha_1, \dots, \alpha_n) \text{ distinct in } \mathbf{F}_{2^m} \\ g(z) \in \mathbf{F}_{2^m}[z] \text{ monic irreducible of degree } t \end{cases}$

**Public key:**  $R \in \mathbf{F}_2^{k \times (n-k)}$

where  $G = (\text{Id} \mid R)$  is a systematic generator matrix of  $\Gamma(L, g)$

**Encryption:**  $\left[ \begin{array}{l} E_R : \mathbf{F}_2^k \times \mathbf{F}_2^\ell \rightarrow \mathbf{F}_2^n \\ (x, x') \mapsto (x, xR) + \varphi(x') \end{array} \right]$

**Decryption:**  $\left[ \begin{array}{l} D_{L,g} : \mathbf{F}_2^n \rightarrow \mathbf{F}_2^k \times \mathbf{F}_2^\ell \\ y \mapsto (x, x') \end{array} \right]$

where  $(x, *) = \Phi_{L,g}(y)$  and  $x' = \varphi^{-1}(y - \Phi_{L,g}(y))$

## Security of Hybrid McEliece

- Using the error for encoding information

No security loss!

In fact, there is a loss of a factor at most  $2^\ell / \binom{n}{t}$

- Using a systematic generator matrix

The system remains OWE, puzzling but true!

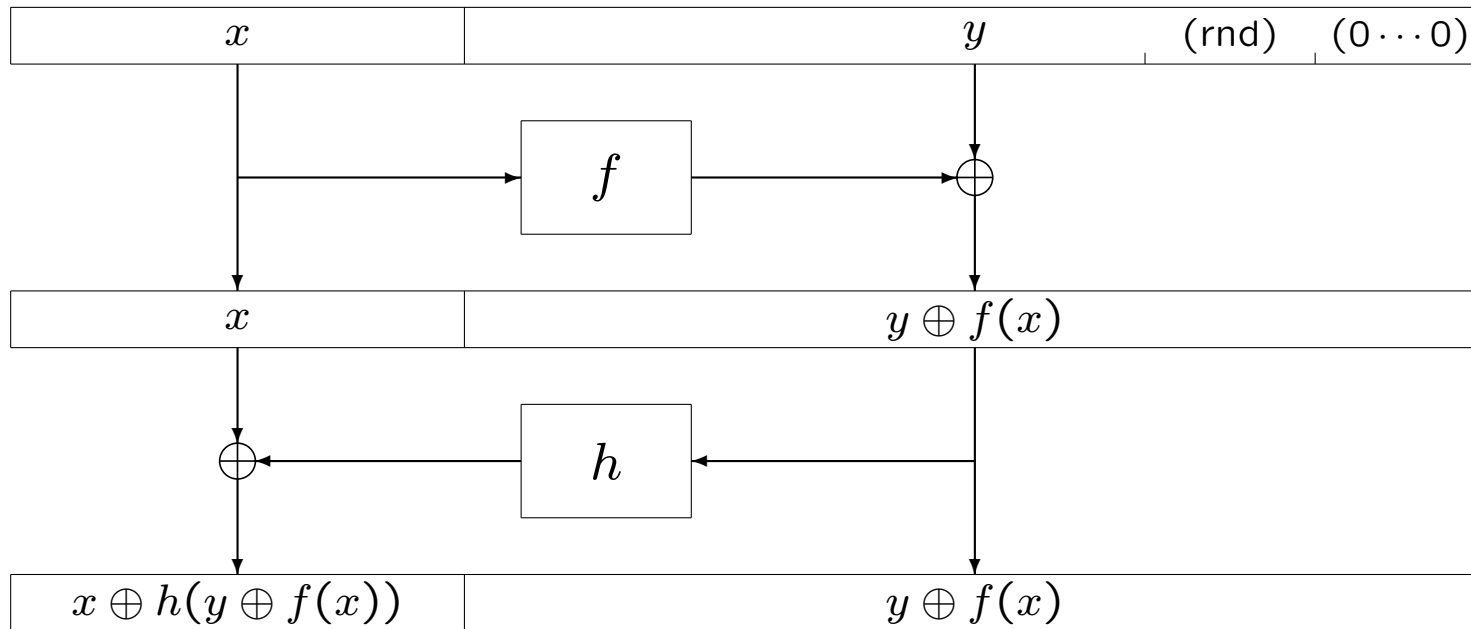
cleartext:  $x$

ciphertext:  $(x, xR) + e$  with  $e$  of small weight

No change in security, but there is a need for a semantically secure layer (as for the original system)

# Conversion for Semantic Security – OAEP

[Bellare & Rogaway, 94]



$$\text{2-round Feistel scheme} \begin{cases} a = x \oplus h(y \oplus f(x)) \\ b = y \oplus f(x) \end{cases} \Leftrightarrow \begin{cases} x = a \oplus h(b) \\ y = b \oplus f(a \oplus h(b)) \end{cases}$$

Under the “random oracle assumption” on  $f$  and  $h$  this conversion provides semantic security (non malleability and indistinguishability).

## Encryption/Decryption Speed

$m, t$	sizes		cycles/byte		cycles/block		security
	cipher	clear	encrypt	decrypt	encrypt	decrypt	
11, 40	2048	1888	105	800	25K	189K	81
12, 50	4096	3881	98	618	47K	300K	120

(Intel Xeon 3.4Ghz, single processor) 100 Kcycle  $\approx 30 \mu s$

AES: 10-20 cycles/byte

**McBits** [Berstein, Chou, & Schwabe] gains a factor  $\approx 5$  on decoding (bit-sliced field arithmetic + algorithmic innovations for decoding).  
Targets key exchange mechanism based on Niederreiter.



# V. Practical Security - The Attacks

## Best Known Attacks

**Decoding attacks.** For the public-key encryption schemes the best attack is always Information Set Decoding (ISD), this will change for other cryptosystems

**Key attacks.** Most proposals using families other than binary Goppa codes have been broken

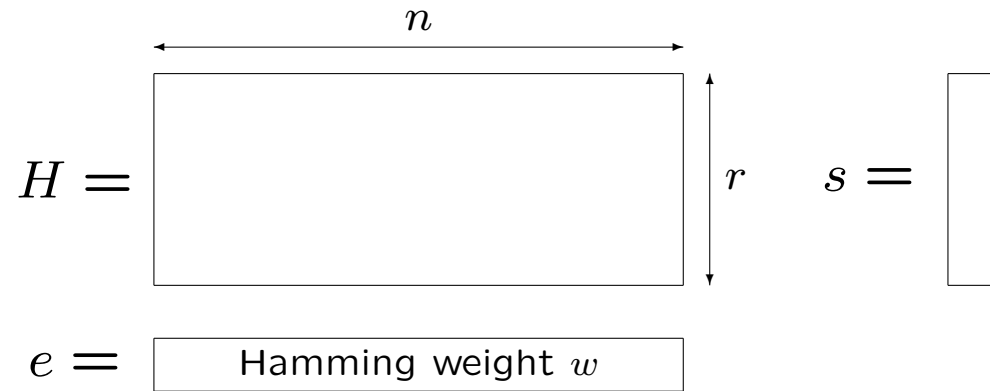
For binary Goppa codes there are only exhaustive attacks enumerating either generator polynomials either supports (that is permutations)

# Syndrome Decoding – Problem Statement

## Computational Syndrome Decoding

CSD( $n, r, w$ )

Given  $H \in \mathbb{F}_2^{r \times n}$  and  $s \in \mathbb{F}_2^r$ , solve  $eH^T = s$  with  $|e| \leq w$

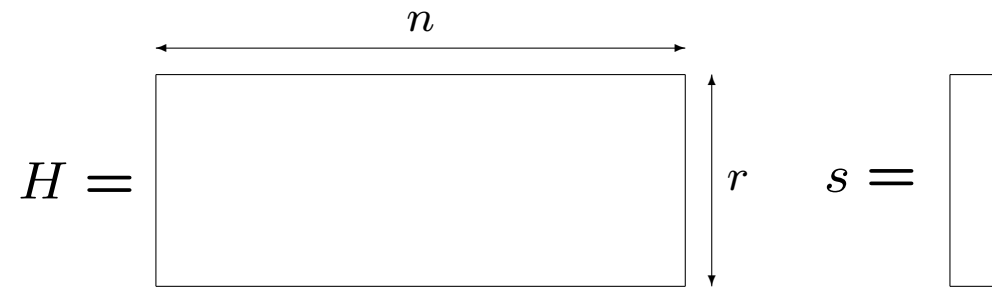


**Find  $w$  columns of  $H$  adding to  $s$**

Very close to a subset sum problem

For instance  $\begin{cases} n = 2048 \\ r = 352 \\ w = 32 \end{cases} \rightarrow$  computing effort  $> 2^{80}$

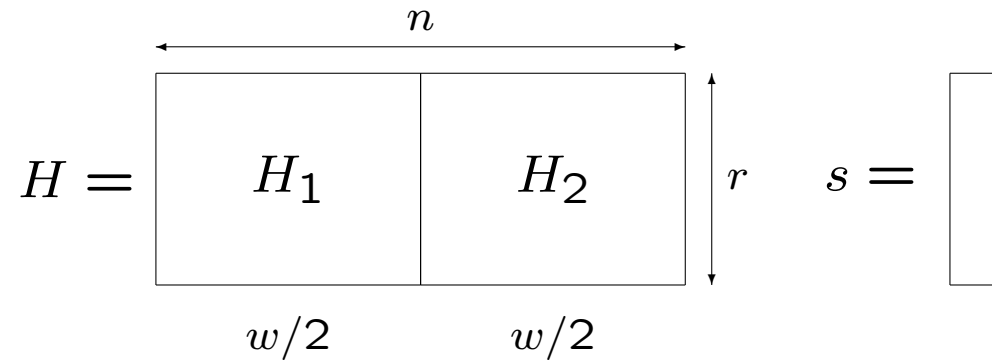
## Algorithm 0



Compute every sum of  $w$  columns  $\rightarrow$  complexity  $\binom{n}{w}$  column ops.

1 column operation  $\left\{ \begin{array}{l} 1 \text{ read or write} \\ \text{and} \\ 1 \text{ test} \\ \text{and} \\ 1 \text{ addition or weight computation} \end{array} \right.$

## Algorithm 1: Birthday Decoding



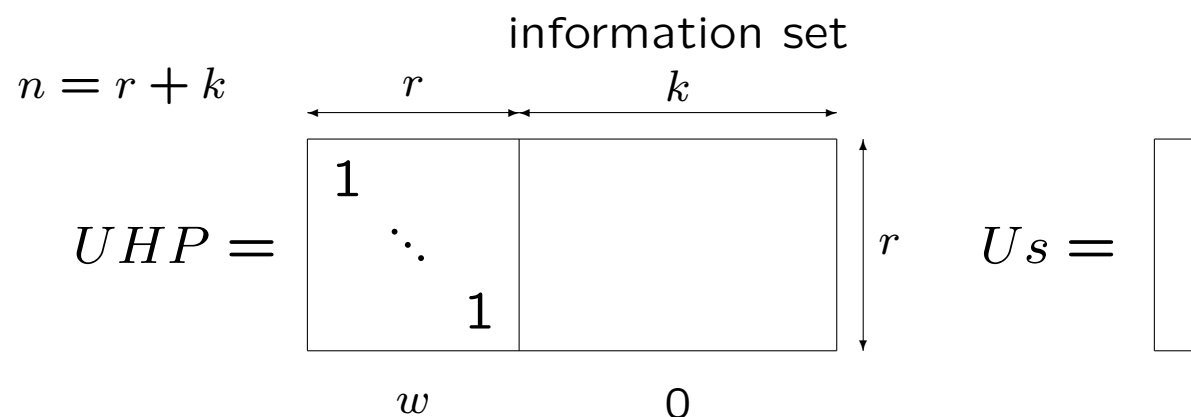
Compute  $\{H_1 e \mid |e| = w/2\} \cap \{s + H_2 e \mid |e| = w/2\}$

Complexity  $2 \binom{n/2}{w/2}$  and non-empty with probability  $\frac{\binom{n/2}{w/2}^2}{\binom{n}{w}}$

→ average cost  $2 \frac{\binom{n}{w}}{\binom{n/2}{w/2}} \approx \sqrt[4]{8\pi w} \sqrt{\binom{n}{w}}$

## Algorithm 2: Information Set Decoding [Prange, 1962]

Big difference with subset sums: one can use linear algebra



Repeat for several permutation matrices  $P$

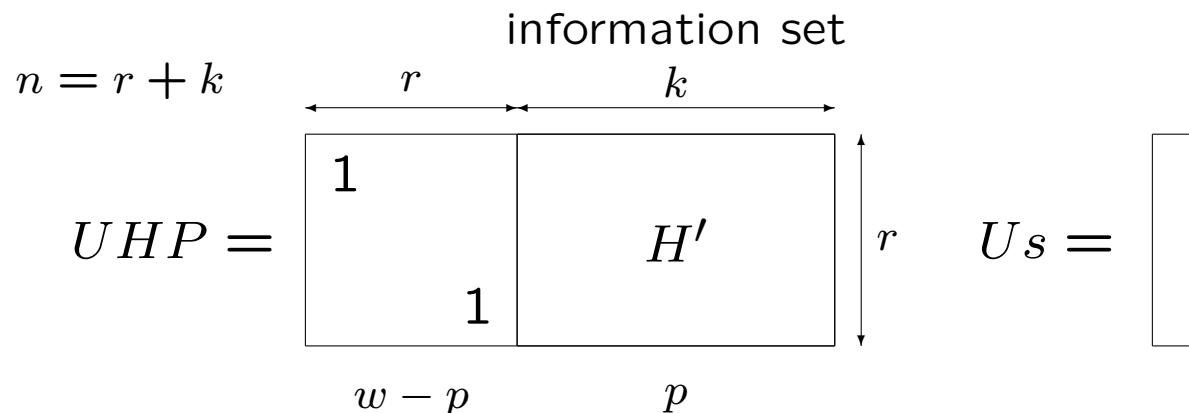
**Claim:** if  $|U_S| \leq w$ , I win!

Success probability:  $\binom{r}{w} / \binom{n}{w} \approx (r/n)^w$

Total cost:  $\approx rn(n/r)^w$  column operations

## Algorithm 2': ISD [Lee & Brickell, 1988]

Idea: amortize the Gaussian elimination



Repeat for several permutation matrices  $P$

**Claim:** if  $\exists e$  with  $|e| = p$  and  $|U_S + H'e| = w - p$ , I win!

Success probability:  $\frac{\binom{r}{w-p} \binom{k}{p}}{\binom{n}{w}}$       Iteration cost:  $rn + \binom{k}{p}$

Total cost:  $\frac{\binom{n}{w}}{\binom{r}{w-p}} \left( 1 + \frac{rn}{\binom{k}{p}} \right)$ , only a polynomial gain

# Generalized Information Set Decoding

[Stern, 89] ; [Dumer, 91]

$$UHP = \begin{array}{|c|c|} \hline 1 & H'' \\ \hline 0 & H' \\ \hline \end{array} \begin{array}{l} \xleftarrow{k + \ell} \\ \uparrow r - \ell \\ \downarrow \ell \\ \xrightarrow{w - p} \quad \xrightarrow{p} \end{array} \quad Us = \begin{array}{|c|} \hline s'' \\ \hline s' \\ \hline \end{array}$$

Repeat:  $\left\{ \begin{array}{l} 1. \text{ Permutation + partial Gaussian elimination} \\ 2. \text{ Find many } e' \text{ of weight } p \text{ such that } H'e' = s' \\ 3. \text{ For all good } e', \text{ test } |s'' + H''e'| \leq w - p \end{array} \right.$

Step 3. is (a kind of) Lee & Brickell which embeds Step 2

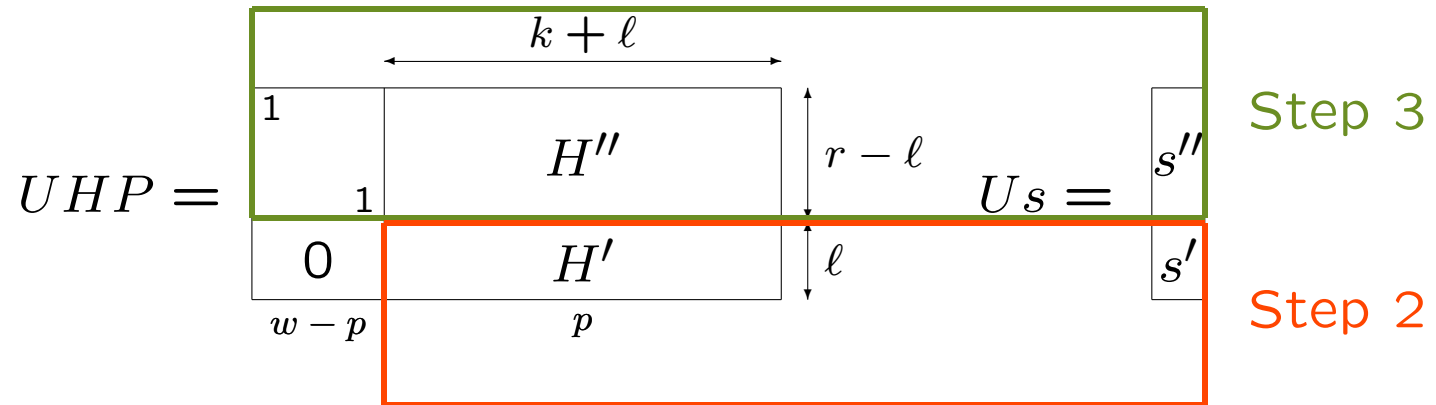
Step 2. is Birthday Decoding (or whatever is best)

Total cost is minimized over  $\ell$  and  $p$



# Generalized Information Set Decoding

[Stern, 89] ; [Dumer, 91]



- Repeat: {
1. Permutation + partial Gaussian elimination
  2. Find many  $e'$  of weight  $p$  such that  $H'e' = s'$
  3. For all good  $e'$ , test  $|s'' + H''e'| \leq w - p$

Step 3. is (a kind of) Lee & Brickell which embeds Step 2

Step 2. is Birthday Decoding (or whatever is best)

Total cost is minimized over  $\ell$  and  $p$

# Generalized Information Set Decoding – Workfactor

$$\begin{array}{c}
 \begin{array}{c}
 \xrightarrow{n} \\
 \begin{array}{|c|c|}
 \hline
 1 & H'' \\
 \hline
 0 & H' \\
 \hline
 \end{array} \\
 \begin{array}{l}
 \uparrow r - \ell \\
 \downarrow \ell \\
 \end{array} \\
 \xleftarrow{k + \ell}
 \end{array}
 \quad
 sU^T = \begin{array}{|c|}
 \hline
 s'' \\
 \hline
 s' \\
 \hline
 \end{array}
 \\
 \\
 eP = \begin{array}{|c|c|}
 \hline
 & e' \\
 \hline
 \end{array} \\
 \begin{array}{cc}
 w - p & p \\
 \leftarrow & \text{weight profile}
 \end{array}
 \end{array}$$

Assuming the Gaussian elimination cost is not significant

$$\text{WF}_{\text{ISD}} = \min_{p, \ell} \frac{\binom{n}{w}}{\binom{r-\ell}{w-p} \binom{k+\ell}{p}} \left( \sqrt{\binom{k+\ell}{p}} + \frac{\binom{k+\ell}{p}}{2^\ell} \right)$$

column operations up to a small constant factor. Simplifies to

$$\text{WF}_{\text{ISD}} = \min_p \frac{\binom{n}{w}}{\binom{r-\ell}{w-p} \sqrt{\binom{k+\ell}{p}}} \text{ with } \ell = \log \left( \sqrt{\binom{k+\ell}{p}} \right)$$

## Information Set Decoding – Timeline

- Information Set Decoding: [Prange, 62]
- Relax the weight profile: [Lee & Brickell, 88]
- Compute sums on partial columns first: [Leon, 88]
- Use the birthday attack: [Stern, 89], [Dumer, 91]
- First “real” implementation: [Canteaut & Chabaud, 98]
- Initial McEliece parameters broken: [Bernstein, Lange, & Peters, 08]
- Lower bounds: [Finiasz & Sendrier, 09]
- Ball-collision decoding [Bernstein, Lange, & Peters, 11]
- Asymptotic exponent improved [May, Meurer, & Thomae, 11]
- Decoding one out of many [Sendrier, 11]
- Even better asymptotic exponent [Becker, Joux, May, & Meurer, 12]
- “Nearest Neighbor” variant [May & Ozerov, 15]
- Sublinear error weight [Canto Torres & Sendrier, 16]

## Key Security

This is the main security issue in code based cryptography

- Find families of codes whose generator matrices are indistinguishable from random matrices
- Goppa codes: excluding a few extremal cases, Goppa codes (binary or not) seem to be pseudorandom → best attack is essentially an exhaustive search

We assume it is true, do we have better arguments?

- Can we find quasi-cyclic families which are indistinguishable?  
QC-MDPC is an answer to some extent. Can we do better?

## Conclusion for Public Key Encryption

- Good security reduction  
partly heuristic though:
  - nothing proven on the average case complexity of decoding
  - indistinguishability assumptions need more attention
- The best attacks are decoding attacks
  - generic decoding is an essential long term research topic (including with quantum algorithms)
- Open problems are mainly related to the key security
  - find other good families of codes
  - safely reduce the public key size

# VI. Other Public Key Systems

## Other Public Key Systems

- Digital Signature, [Courtois, Finiasz & Sendrier, 01]  
Same kind security reduction:  
Hardness of decoding & Indistinguishability of Goppa codes
- Zero Knowledge identification  
[Stern, 93], [Véron, 95], [Gaborit & Girault, 07]  
Much stronger security reduction: Hardness of decoding only
- And also...  
ID based signature [Cayrel, Gaborit & Girault, 07]  
Threshold ring signature [Aguilar-Melchor, Cayrel & Gaborit, 08],

## CFS Digital Signature

$H \in \mathbf{F}_2^{r \times n}$  a parity check matrix of a  $t$ -error correcting Goppa code

Signing: the message  $M$  is given

- Hash the text  $M$  into a binary word  $h(M) = s \in \mathbf{F}_2^r$
- Find  $e$  of minimal weight such that  $eH^T = s$
- Use  $e$  as a signature

Verifying:  $M$  and  $e$  are given

- Hash the text  $M$  into a binary word  $h(M) = s \in \mathbf{F}_2^r$
- Check  $eH^T = s$



## CFS Digital Signature – Not so Easy

In practice  $n = 2^m = 2^{16}$ ,  $t = 9$  and  $r = n - k = tm = 144$

The public key  $H$  has size  $144 \times 65536$  ( $\approx 1.2$  MB)

Let  $s \in_R \mathbb{F}_2^{144}$ , let  $w$  be the minimal weight of  $e$  such that  $s = eH^T$

- $w \leq 9$  with probability  $\approx 3 \cdot 10^{-6}$  (in general  $w \leq t$  with prob.  $1/t!$ )
- $w = 10$  with probability  $\approx 10^{-2}$
- $w = 11$  with probability  $\approx 1 - 10^{-46}$

$w = 11$  is the smallest number such that  $\binom{2^{16}}{11} > 2^{144}$

Problem:

- the trapdoor only allows the correction of  $t = 9$  errors
- we need to decode 11 errors  $\rightarrow$  we have to guess 2 error positions
- requires  $t! = 362880$  decoding attempts on average

The legitimate user has to pay  $\approx 2^{33}$  while the attacker has to pay  $> 2^{77}$

## CFS Digital Signature – Scalability

Binary Goppa code of length  $n = 2^m$  correcting  $t$  errors

The public key  $H \in \mathbb{F}_2^{r \times n}$  (where  $r = tm$  is the codimension)

Signature cost	$t!O(m^2t^2)$
Signature length	$tm - \log_2(t!)$
Verification cost	$O(mt^2)$
Public key size	$tm2^m$
Security bits	$\frac{1}{2}tm$

- The signature cost is exponential in  $t$
- The key size is exponential in  $m$
- The security is exponential in  $tm$

## CFS Digital Signature – Decoding One Out of Many

Bleichenbacher's "Decoding One Out of Many"-type attack (2003 or 2004, unpublished) reduces the security to  $\frac{1}{3}tm$

[Finiasz, 10] Parallel-CFS: sign several related syndrome.

- take a ( $\lambda$  times) longer hash of the message  $h(M) = (s_1, \dots, s_\lambda)$
- sign all  $\lambda$  syndromes  $\rightarrow$  security back to  $\frac{1}{2}tm$
- $\lambda$  must be 3 or 4 (do not need to grow with the security parameter)

Signature length & cost and verification cost all multiplied by  $\lambda$

## CFS Digital Signature – Implementation

- [Landais & Sendrier, 12] Software implementation of parallel-CFS  
 $(m, t) = (20, 8)$ ,  $\lambda = 3 \rightarrow 80$  bits security  
Key size: 20 MB, one signature in  $\approx 1.5$  seconds
- [+ Schwabe] bit-sliced field arithmetic  $\rightarrow$  100 milliseconds for one signature

An important security issue: binary Goppa codes of rate  $\rightarrow 1$  are not pseudorandom (no attack, but no security reduction either)

## Stern ZK Authentication Protocol

Parameters:  $H \in \mathbb{F}_2^{r \times n}$ , weight  $w > 0$ , commitment scheme  $c(\cdot)$

Secret: some word  $e$  of weight  $w$  ( $w \approx$  Gilbert-Varshamov distance)

Public: the syndrome  $s = eH^T$

	Prover	Verifier
<b>Commitment</b>	$\sigma \leftarrow \mathcal{S}_n$ $y \leftarrow \mathbb{F}_2^n$	$\xrightarrow{c_0, c_1, c_2}$
<b>Challenge</b>	$\xleftarrow{b}$	$b \leftarrow \{0, 1, 2\}$
<b>Answer</b>	$\xrightarrow{A_b}$	check commitments

$$\begin{cases} c_0 = c(\sigma(y + e)) \\ c_1 = c(yH^T, \sigma) \\ c_2 = c(\sigma(y)) \end{cases}
 \quad
 \begin{cases} A_0 = y, \sigma \\ A_1 = \sigma(y), \sigma(e) \\ A_2 = (y + e), \sigma \end{cases}$$

$$\text{Check: } \begin{cases} \text{if } b = 0 \text{ check } c_1 \text{ and } c_2 \\ \text{if } b = 1 \text{ check } c_0 \text{ and } c_2 \text{ (and } |\sigma(e)| = w) \\ \text{if } b = 2 \text{ check } c_0 \text{ and } c_1 \end{cases}$$

## Stern ZK Authentication Protocol – Security

- An honest prover always succeeds (**completeness**)
  - A dishonest prover succeeds for one round with probability  $2/3$  at most (eventually leading to **soundness**)
  - No information on the secret leaks (**zero-knowledge**)
- For a security level  $S$ ,  $S/\log_2(3/2) \approx 1.7S$  rounds are needed  
(80 bits security → 137 rounds, 128 bits security → 219 rounds)
- Can be transformed into a signature (Fiat-Shamir NIZK)
- A tight security reduction to syndrome decoding

## Signing with Stern ZK Protocol

	Prover	Verifier
<b>Commitment</b>	$\sigma_i \leftarrow \mathcal{S}_n$ $y_i \leftarrow \mathbf{F}_2^n$	$c_{0,i}, c_{1,i}, c_{2,i}$ $\xrightarrow{\quad}$
<b>Challenge</b>	$\xleftarrow{\quad} b_i$	$b_i \leftarrow \{0, 1, 2\}$
<b>Answer</b>	$\xrightarrow{\quad} A_{b_i,i}$	check commitments

- Draw  $\sigma_i, y_i$ , and compute  $c_{0,i}, c_{1,i}, c_{2,i}$  for all  $i, 1 \leq i \leq R$
- Compute  $x = Hash((c_{0,i}, c_{1,i}, c_{2,i})_{1 \leq i \leq R})$
- Draw  $b_i, 1 \leq i \leq R$ , using a PRNG with seed  $x$
- The signature is  $(A_{b_i,i}, c_{0,i}, c_{1,i}, c_{2,i})_{1 \leq i \leq R}$

80 bits security  $\rightarrow$  signature of 174 Kbits

128 bits security  $\rightarrow$  signature of 445 Kbits

[Aguilar-Melchor, Gaborit, & Schrek, 11] reduced to 79 and 202 Kbits

## General Conclusions

- Code-based cryptosystems are practical, efficient, secure, versatile  
... some of them at least
- Also symmetric schemes (hash function, stream ciphers, . . . )
- Strong features
  - Hardness of decoding, tight security reductions in that respect
  - Efficient algorithms: fast public key encryption
- Not so strong features
  - Public key size (not necessarily a problem)
  - Few code families: biodiversity would be welcome
- Main open problems
  - Key security (security assumptions, families of codes, . . . )
  - Key size reduction: what gain for what cost?
  - Improve the digital signature



Thank you for your attention

Appendix

—

MDPC McEliece

## QC-MDPC-McEliece Scheme (1/2)

Parameters:  $n, k, w, t$

(for instance  $n = 9601, k = 4801, w = 90, t = 84$ )

**Key generation:** (rate  $1/2, n = 2p, k = p$ )

Pick a (sparse) vector  $(h_0, h_1) \in \mathbb{F}_2^p \times \mathbb{F}_2^p$  of weight  $w$

$$H_{\text{secret}} = \begin{array}{|c|c|} \hline \boxed{h_0} & \boxed{h_1} \\ \hline \circlearrowleft & \circlearrowleft \\ \hline \end{array}$$

with  $h_0(x)$  invertible in  $\mathbb{F}_2[x]/(x^p - 1)$

(circulant binary  $p \times p$  matrices are isomorphic to  $\mathbb{F}_2[x]/(x^p - 1)$ )

Publish  $h(x) = h_1(x)h_0^{-1}(x) \pmod{x^p - 1}$  or  $g(x) = \overline{h(x)}/x$

$$H = \begin{array}{|c|c|} \hline \mathbf{1} & \boxed{h} \\ \hline \backslash & \circlearrowleft \\ \hline & \mathbf{1} \\ \hline \end{array} \quad \text{or} \quad G = \begin{array}{|c|c|} \hline \boxed{g} & \mathbf{1} \\ \hline \circlearrowleft & \backslash \\ \hline & \mathbf{1} \\ \hline \end{array}$$

$H$  a parity check matrix,  $G$  a generator matrix

## QC-MDPC-McEliece Scheme (2/2)

**Encryption:** (rate  $1/2$ ,  $n = 2p$ ,  $k = p$ )

$$\begin{aligned}\mathbf{F}_2[x]/(x^p - 1) &\rightarrow \mathbf{F}_2[x]/(x^p - 1) \times \mathbf{F}_2[x]/(x^p - 1) \\ m(x) &\mapsto (m(x)g(x) + e_0(x), m(x) + e_1(x))\end{aligned}$$

The error  $e(x) = (e_0(x), e_1(x))$  has weight  $t$

**Decryption:**

Iterative decoding (as for LDPC codes) which only requires the sparse parity check matrix. For instance the “bit flipping” algorithm

Parameters are chosen such that the decoder fails to correct  $t$  errors with negligible probability

Each iteration has a cost proportional to  $w \cdot (n - k)$ , the number of iterations is small (3 to 5 in practice)

## QC-MDPC-McEliece Security Reduction

$$H = \left[ \begin{array}{c|c} 1 & \boxed{h} \\ \backslash & \circlearrowleft \\ & 1 \end{array} \right] \text{ with } h(x) = \frac{h_1(x)}{h_0(x)} \pmod{x^p - 1}$$

Secure under two assumptions

1. Pseudorandomness of the public key

Hard to decide whether there exists a sparse vector in the code spanned by  $H$  (the dual of the MDPC code)

2. Hardness of generic decoding of QC codes

Hard to decode in the code of parity check matrix  $H$  (for an arbitrary value of  $h$ )

## QC-MDPC — Sparse Polynomial Problems

The security reduction and the attacks can be stated in terms of polynomials

### 1. Key Security

Given  $h(x)$ , find non-zero  $(h_0(x), h_1(x))$  such that

$$\begin{cases} h_0(x) + h(x)h_1(x) = 0 \pmod{x^p - 1} \\ |h_0| + |h_1| \leq w \end{cases}$$

or simply decide the existence of a solution  $\rightarrow$  distinguisher

### 2. Message Security

Given  $h(x)$  and  $S(x)$ , find  $e_0(x)$  and  $e_1(x)$  such that

$$\begin{cases} e_0(x) + h(x)e_1(x) = S(x) \pmod{x^p - 1} \\ |e_0| + |e_1| \leq t \end{cases}$$

In both cases, best known solutions use generic decoding algorithms

## QC-MDPC — Practical Security – Best Known Attacks

Let  $W_{SD}(n, k, t)$  denote the cost for the generic decoding of  $t$  errors in a binary  $[n, k]$  code

We consider a QC-MDPC-McEliece instance with parameters  $n, k, w, t$  and circulant blocks of size  $p$ .

1. **Key Attack:** find a word of weight  $w$  in a quasi-cyclic binary  $[n, n - k]$  code

$$W_K(n, k, w) \geq \frac{W_{SD}(n, n - k, w)}{n - k}$$

(there are  $n - k$  words of weight  $w$ )

2. **Message Attack:** decode  $t$  errors in a quasi-cyclic binary  $[n, k]$  code

$$W_M(n, k, t, p) \geq \frac{W_{SD}(n, k, t)}{\sqrt{p}}$$

(Decoding One Out of Many [S., 11]  $\rightarrow$  factor  $\sqrt{p}$ )

## QC-MDPC — Parameter Selection

Choose a code rate  $k/n$  and a security exponent  $S$  (for instance 80 or 128). Then increase the block size until the following succeeds:

- find  $w$  the smallest integer such that  $W_K(n, k, w) \geq 2^S$
- find  $t$  the error correcting capability of the corresponding MDPC code
- check that  $W_M(n, k, t, p) \geq 2^S$

80 bits of security

$$n = 9602$$

$$k = 4801$$

$$p = 4801$$

$$w = 90$$

$$t = 84$$

128 bits of security

$$n = 19714$$

$$k = 9857$$

$$p = 9857$$

$$w = 142$$

$$t = 134$$



## QC-MDPC — Scalability

A binary  $[n, k]$  code with  $n - k$  parity equations of weight  $w$  will correct  $t$  errors with an LDPC-like decoding algorithm as long as  $t \cdot w \lesssim n$

For LDPC codes, we have essentially  $w = O(1)$ . For MDPC codes we have  $w = O(\sqrt{n})$  and thus  $t = O(\sqrt{n})$ .

The optimal trade-off between the key size ( $K$ ) and the security ( $S$ ) is obtained for codes of rate  $1/2$  and

$$K \approx cS^2 \text{ with } c < 1$$

For Goppa code, the optimal code rate is  $\approx 0.8$  and

$$K \approx c(S \log_2 S)^2 \text{ with } c \approx 2$$

## QC-MDPC — Bit-Flipping Decoding

Parameter: a threshold  $T$

input:  $y \in \mathbf{F}_2^n$ ,  $H \in \mathbf{F}_2^{(n-k) \times n}$

Repeat

    Compute the syndrome  $Hy^T$

    for  $j = 1, \dots, n$

        if more than  $T$  parity equations involving  $j$  are violated then  
            flip  $y_j$

$$Hy^T = \begin{pmatrix} s_1 \\ \vdots \\ s_{n-k} \end{pmatrix}, \text{ if } s_i \neq 0 \text{ the } i\text{-th parity equation is violated}$$

If  $H$  is sparse enough and  $y$  close to the code of parity check matrix  $H$  then the algorithm finds the closest codeword after a few iterations