Code-Based Public-Key Cryptography

Nicolas Sendrier

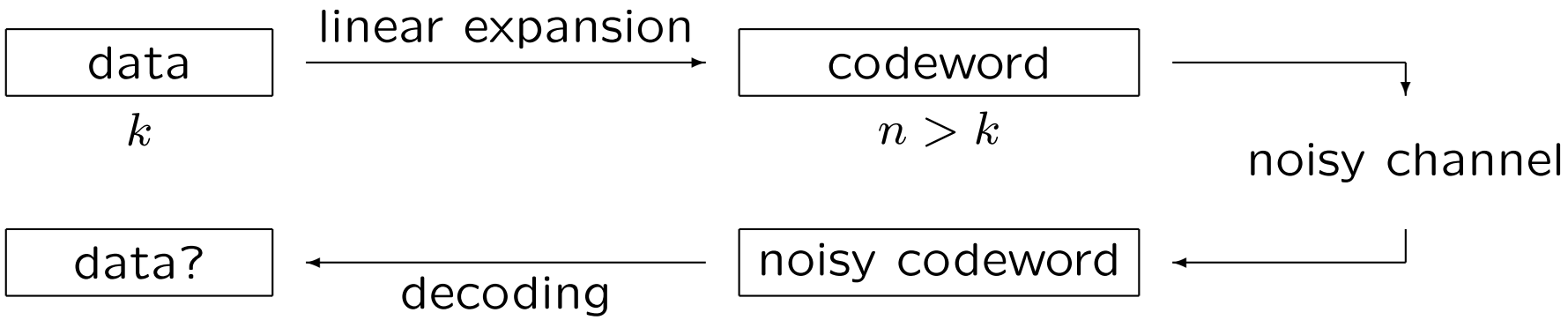
Code-Based Public-Key Cryptography

Post-Quantum Cryptography Summer School
Waterloo, Canada, September 30, 2014

Nicolas Sendrier

informatics mathematics

Linear Codes for Telecommunication



[Shannon, 1948] (for a binary symmetric channel of error rate p):
Decoding probability $\longrightarrow 1$ 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 λn errors ($\lambda = h^{-1}(1-R)$)

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

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

Random Codes Are Hard to Decode

When the linear expansion is random:

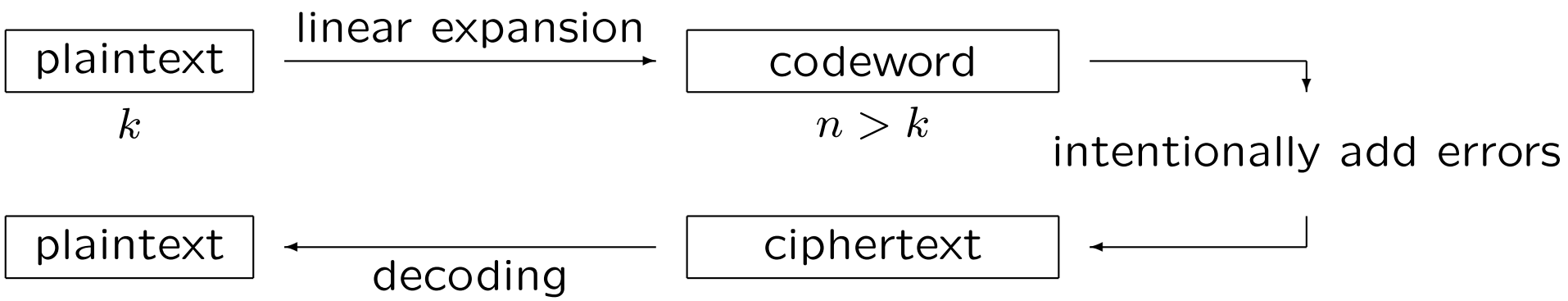
- Decoding is NP-complete [Berlekamp, McEliece & van Tilborg, 78]
- Even the tiniest amount of error is (believed to be) hard to remove. Decoding n^{ε} errors is conjectured difficult on average for any $\varepsilon > 0$ [Alekhnovich, 2003].

Codes with Good Decoders Exist

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

- alternant codes have a poly-time decoder for $O\left(\frac{n}{\log n}\right)$ errors
- some classes of codes have a poly-time decoder for $O(n)$ 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

  $\rightarrow$ that's what this lecture is 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: $x = (x_1, \ldots, x_n) \in \mathbf{F}_q^n$, $y = (y_1, \ldots, y_n) \in \mathbf{F}_q^n$

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

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

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

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

(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 $\mathcal{C} = \left\{ xG \mid x \in \mathbf{F}_q^k \right\}$
It defines an encoder for $\mathcal{C}$

$$\begin{array}{rccc} f_G : & \mathbf{F}_q^k & \to & \mathcal{C} \\ & x & \mapsto & xG \end{array}$$

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

If G is in systematic form, $G = (\mathrm{Id} \mid R)$ then $G^* = (\mathrm{Id} \mid \mathbf{0})^T$ 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 $\mathcal{C} = \left\{x \in \mathbf{F}_q^n \mid xH^T = 0\right\}$

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 $\mathcal{C}$

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

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

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

$$\mathrm{wt}(e) \leq t \Rightarrow \Psi_H(eH^T) = e$$

$$\begin{array}{c} \exists \text{ an efficient} \\ t\text{-bounded decoder} \end{array} \Leftrightarrow \begin{array}{c} \exists \text{ an efficient} \\ t\text{-bounded syndrome decoder} \end{array}$$

McEliece Public-key Encryption Scheme – Overview

Let $\mathcal{C}$ be a binary linear $[n, k]$ code

Public key:	a generator matrix $G \in \{0,1\}^{k \times n}$ of $\mathcal{C}$ $\mathcal{C} = \left\{ xG \mid x \in \{0,1\}^k \right\}$
Secret key:	a t-bounded decoder $\Phi : \{0,1\}^n \to \mathcal{C}$ for $\mathcal{C}$
Encryption:	$\left[\begin{array}{rcl} E_G : \{0,1\}^k & \to & \{0,1\}^n \\ x & \mapsto & xG + e \end{array} \right]$ with e random of weight t
Decryption:	$\left[\begin{array}{rcl} D_\Phi : \{0,1\}^n & \to & \{0,1\}^k \\ y & \mapsto & \Phi(y)G^* \end{array} \right]$ 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 $\mathcal{C}$ be a binary linear $[n,k]$ code, $r = n - k$

Public key: a parity check matrix $H \in \{0,1\}^{r \times n}$ of $\mathcal{C}$

$$\mathcal{C} = \left\{ x \in \{0,1\}^n \mid xH^T = 0 \right\}$$

Secret key: a t-bounded H-syndrome decoder $\Psi : \{0,1\}^r \to \{0,1\}^n$

Encryption:

$$\left[\begin{array}{rccc} E_H : & \mathcal{S}_n(\mathbf{0}, t) & \to & \{0,1\}^r \\ & e & \mapsto & eH^T \end{array} \right]$$

Decryption:

$$\left[\begin{array}{rccc} D_\Psi : & \{0,1\}^r & \to & \mathcal{S}_n(\mathbf{0}, t) \\ & s & \mapsto & \Psi(s) \end{array} \right]$$

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 binary $[n, k]$ 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 was an irreducible binary Goppa code of length 1024, dimension 524 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)	S. (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 $\to 1$

More on Goppa Codes

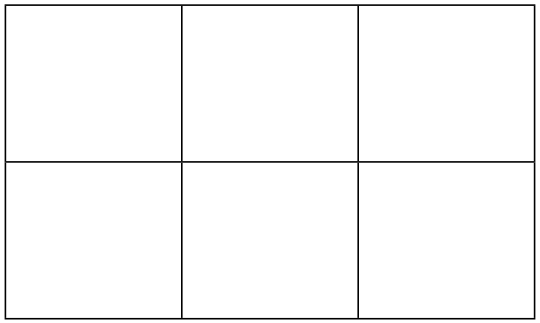
Goppa codes are not limited to the binary case. It is possible to define q-ary Goppa codes with a support in $\mathbf{F}_{q^m}$.

[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



- 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, S., & Barreto, 13]. As above with a stronger security reduction.

Irreducible Binary Goppa Codes

Parameters: m, t and $n \leq 2^m$
Support: $L = (\alpha_1, \ldots, \alpha_n)$ distinct in $\mathbf{F}_{2^m}$
Generator: $g(z) \in \mathbf{F}_{2^m}[z]$ monic irreducible of degree t

For all $a = (a_{\alpha_1}, \ldots, a_{\alpha_n}) \in \mathbf{F}_2^n$ (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

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

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

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

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

Some Sets of Parameters for Goppa Codes

	text size in bits					
	McEliece		Niederreiter		key	message
m, t	cipher	clear	cipher	clear	size	security*
$10, 50$	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

	size in bits			security*	
(n, k, w, t)	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: $H \in \{0,1\}^{r \times n}$, $s \in \{0,1\}^r$, w integer

Question: Is there $e \in \{0,1\}^n$ such that $\mathrm{wt}(e) \leq w$ and $eH^T = s$?

Computational Syndrome Decoding NP-hard

Instance: $H \in \{0,1\}^{r \times n}$, $s \in \{0,1\}^r$, w integer

Output: $e \in \{0,1\}^n$ such that $\mathrm{wt}(e) \leq w$ and $eH^T = s$

[Finiasz, 04]

Goppa Bounded Decoding NP-hard

Instance: $H \in \{0,1\}^{r \times n}$, $s \in \{0,1\}^r$

Output: $e \in \{0,1\}^n$ such that $\mathrm{wt}(e) \leq \dfrac{r}{\log_2 n}$ and $eH^T = s$

Open problem: average case complexity (Conjectured difficult)

Hard Structural Problems

Goppa code Distinguishing NP

Instance: $G \in \{0,1\}^{k \times n}$

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 $\to 1$
  (Faugère, Gauthier, Otmani, Perret, & Tillich, 11)

Goppa code Reconstruction

Instance: $G \in \{0,1\}^{k \times n}$

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

- Tightness: gap between decisional and computational problems

Decoders and Distinguishers

For given parameters n, k, and t
Let $\mathcal{G} \subset \mathcal{K} \subset \{0,1\}^{k \times n}$, 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 $\mathcal{K} = \{0,1\}^{k \times n}$)

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

We consider two programs

- a decoding algorithm: $\mathcal{A} : \{0,1\}^n \times \{0,1\}^{k \times n} \to \mathcal{S}_n(\mathbf{0}, t)$
- a distinguisher: $\mathcal{D} : \{0,1\}^{k \times n} \to \{\text{true}, \text{false}\}$

We consider the sample space $\Omega = \{0,1\}^k \times \{0,1\}^{k \times n} \times \mathcal{S}_n(\mathbf{0}, t)$ 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	$\mathcal{A} : \{0,1\}^n \times \{0,1\}^{k \times n} \to \mathcal{S}_n(\mathbf{0}, t)$
$\mathcal{G}$ the (real) public key space	$\mathcal{D} : \{0,1\}^{k \times n} \to \{\text{true}, \text{false}\}$

—

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

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

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

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

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

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

Security Reduction for McEliece

Theorem

If there exists a (T, ε)-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):

$\mathcal{D}(G)$:

$x \leftarrow \{0,1\}^k$; $e \leftarrow \mathcal{S}_n(\mathbf{0}, t)$ *// 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 the all attacks are intractable on average when the messages and the keys are uniformly distributed

Loosely speaking, there is no (T, ε)-adversary with T/ε 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 attack

[folklore]

You intercept a ciphertext y corresponding to an unknown message x (*i.e.* $y = xG + e$)

You choose a codeword a and you transmit $y + a$ which is a valid ciphertext for some unknown cleartext different from x

This is not a desirable feature, *a priori...*

Resend-message Attack

[Berson, 97]

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

Adding the two ciphertexts $y_1 = xG + e_1$ and $y_2 = xG + e_2$ we obtain $y_1 + y_2 = e_1 + e_2$

The word $e_1 + e_2$ will have a weight $\rho = 2(t - \nu)$ where ν is the number of overlapping non-zero positions in e_1 and e_2

In practice ν is small (2.5 on average with the original McEliece parameters) and we know all but ν of the error positions in the ciphertexts

Removing the ν remaining errors is a simple matter

Reaction Attack

[Kobara & Imai, 00] ??

In this attack, we assume the system can be used as an oracle in the following sense:

- If the system receives a word at distance $> t$ from the code it answers "INVALID CIPHERTEXT"
- If the system receives a word at distance $\leq t$ from the code it behaves otherwise (for instance, it proceeds with the protocol)

Given a ciphertext y we transform it into a word y' by flipping the i-th bit. If i was an error position y' is at distance $t-1$ from the code, else it is at distance $t+1$. We submit y' and from the answer we know whether or not i was an error position.

We try this for every position and we retrieve the error pattern

In fact this is a proof that there is no gap between "Decisional Syndrome Decoding" and "Computational Syndrome Decoding"

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-CCA2, the strongest security notion.

McEliece is not deterministic but IND-CCA2 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:

$$\begin{array}{rccc} E_H : & \mathcal{S}_n(\mathbf{0}, t) & \to & \{0,1\}^r \\ & e & \mapsto & eH^T \end{array}$$

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

$$\varphi : \{0,1\}^\ell \to \mathcal{S}_n(\mathbf{0}, t)$$

with $\ell < \log_2 \binom{n}{t}$ but as close as possible. In addition, we need φ and φ^{-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 φ 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 $0 \leq i_1 < i_2 < \ldots < i_t < n$ and we define the one-to-one mapping

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

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

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

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

The cost in quadratic in ℓ

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 $\rightarrow$ variable length encoding
- fast $\rightarrow$ 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 & S., 08]

Parameters: m, t, $n = 2^m$, $\varphi : \{0,1\}^\ell \to \mathcal{S}_n(\mathbf{0}, t)$

Secret key: an irreducible binary Goppa code $\Gamma(L, g)$
$\Phi_{L,g}$ a t-bounded decoder

Public key: a systematic generator matrix $G = (\mathrm{Id} \mid R)$ of $\Gamma(L, g)$

Encryption:
$$\left[\begin{array}{rccc} E_R : & \{0,1\}^k \times \{0,1\}^\ell & \to & \{0,1\}^n \\ & (x, x') & \mapsto & (x, xR) + \varphi(x') \end{array}\right]$$

Decryption:
$$\left[\begin{array}{rccc} D_{L,g} : & \{0,1\}^n & \to & \{0,1\}^k \times \{0,1\}^\ell \\ & y & \mapsto & (x, x') \end{array}\right]$$
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

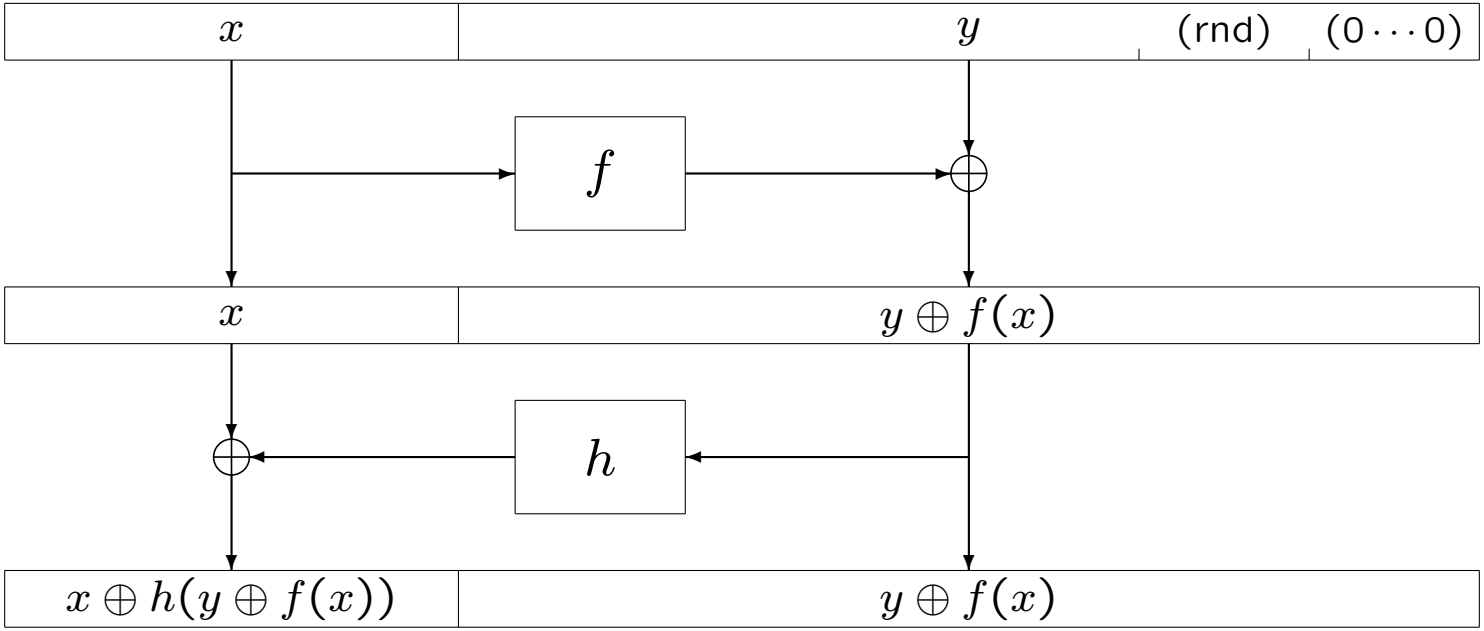
  The system remains OWE, puzzling but true!

  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]



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

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

(Intel Xeon 3.4Ghz, single processor) 100 Kcycle ≈ 30 μs
AES: 10-20 cycles/byte

McBits [Berstein, Chou, & Schwabe] gains a factor ≈ 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

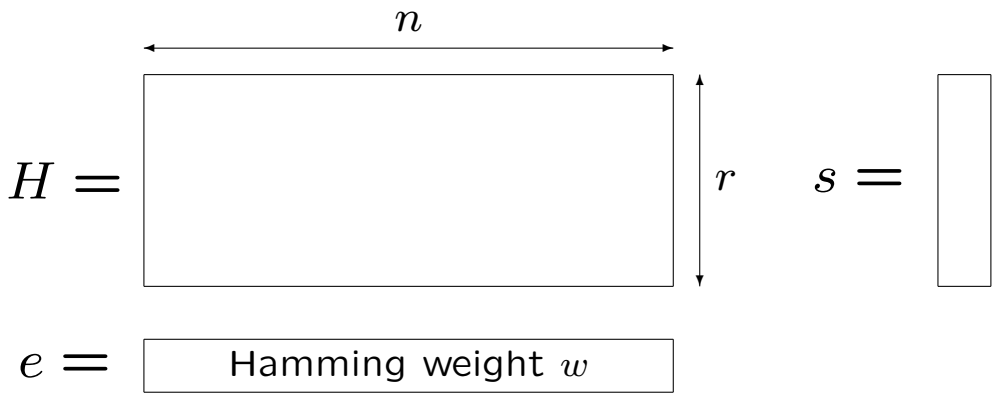
Key attacks. Most proposals using families other than binary Goppa codes have been broken

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 $H \in \{0,1\}^{r \times n}$ and $s \in \{0,1\}^r$, solve $eH^T = s$ with wt$(e) \leq w$

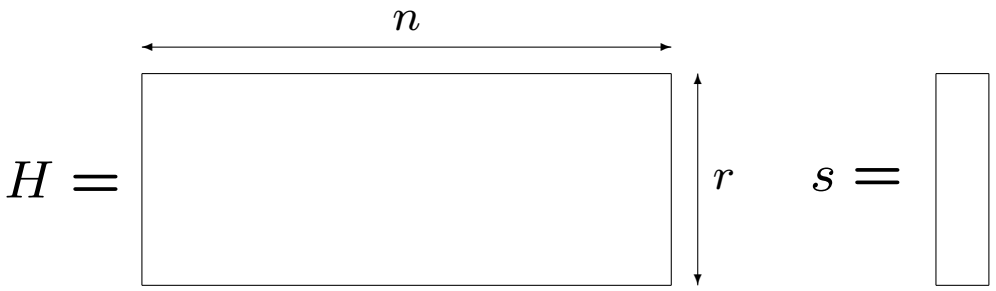


Find w columns of H adding to s

Very close to a subset sum problem

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

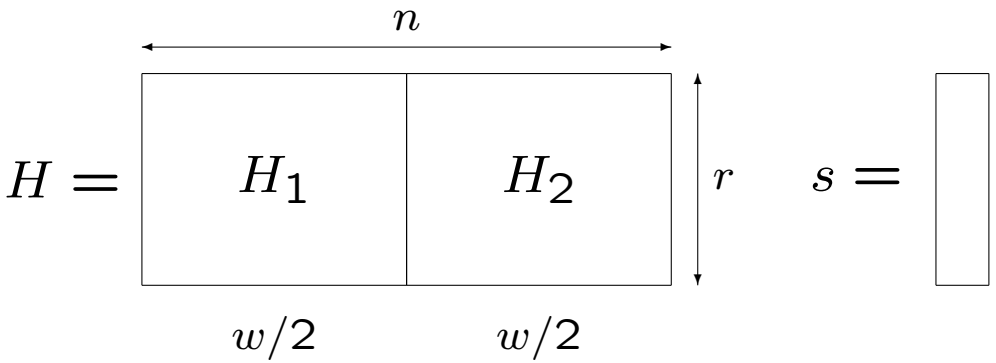
Algorithm 0



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

1 column operation $\begin{cases} \text{1 read or write} \\ \quad \text{and} \\ \text{1 test} \\ \quad \text{and} \\ \text{1 addition or weight computation} \end{cases}$

Algorithm 1: Birthday Decoding



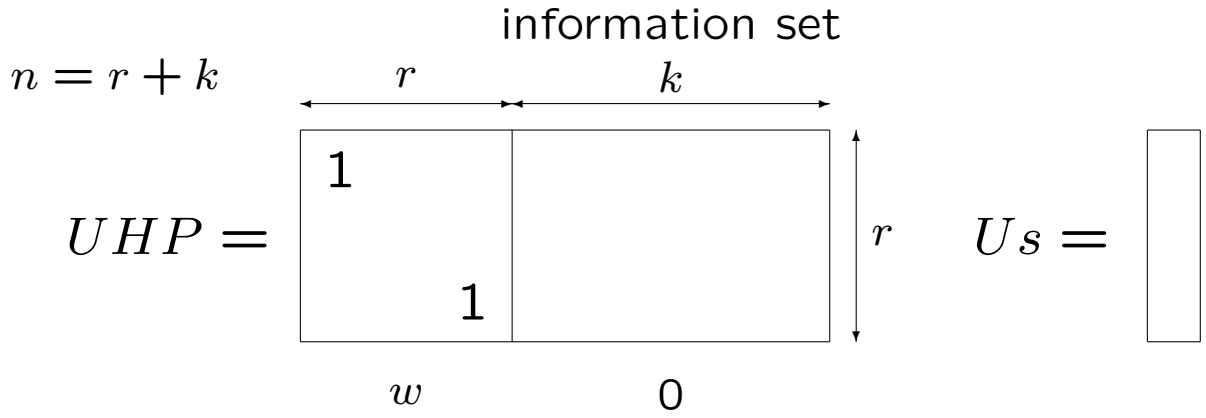
Compute $\{H_1 e \mid \mathrm{wt}(e) = w/2\} \cap \{s + H_2 e \mid \mathrm{wt}(e) = w/2\}$

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

$\rightarrow$ average cost $2\dfrac{\binom{n}{w}}{\binom{n/2}{w/2}} \approx 2\dfrac{\sqrt{\binom{n}{w}}}{\sqrt[4]{w\pi/2}}$ can be reduced to $O\left(\sqrt{\binom{n}{w}}\right)$

Algorithm 2: Information Set Decoding [Prange, 1962]

Big difference with subset sums: one can use linear algebra



Repeat for several permutation matrices P

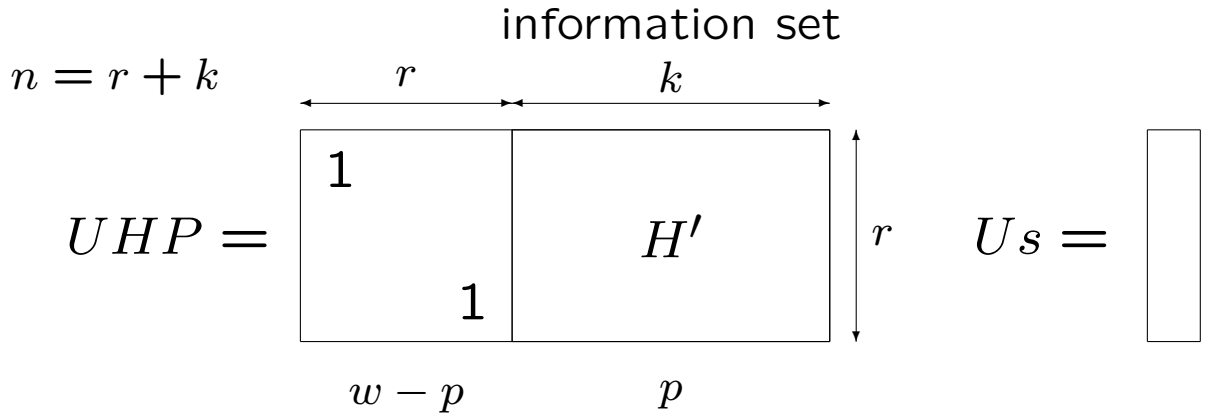
Claim: if $\mathrm{wt}(Us) \leq w$, I win!

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 $\mathrm{wt}(e) = p$ and $\mathrm{wt}(Us + H'e) = w - p$, I win!

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

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

Generalized Information Set Decoding

[Stern, 89] ; [Dumer, 91]

$$UHP = \begin{array}{|cc|c|} \multicolumn{2}{c}{} & \multicolumn{1}{c}{k+\ell} \\ \hline 1 & & \multirow{2}{*}{H''} \\ & 1 & \\ \hline 0 & & H' \\ \hline \multicolumn{1}{c}{w-p} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{p} \end{array} \begin{array}{l} \\ r-\ell \\ \ell \end{array} \qquad Us = \begin{array}{|c|} \hline s'' \\ \hline s' \\ \hline \end{array}$$

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

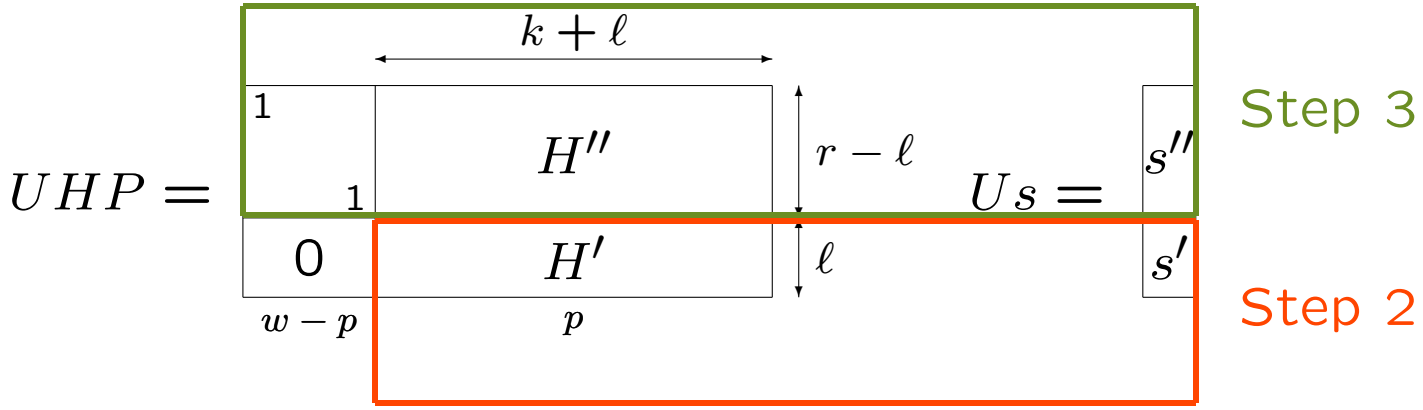
Step 3. is (a kind of) Lee & Brickell which embeds Step 2

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

Total cost is minimized over ℓ and p

Generalized Information Set Decoding

[Stern, 89] ; [Dumer, 91]



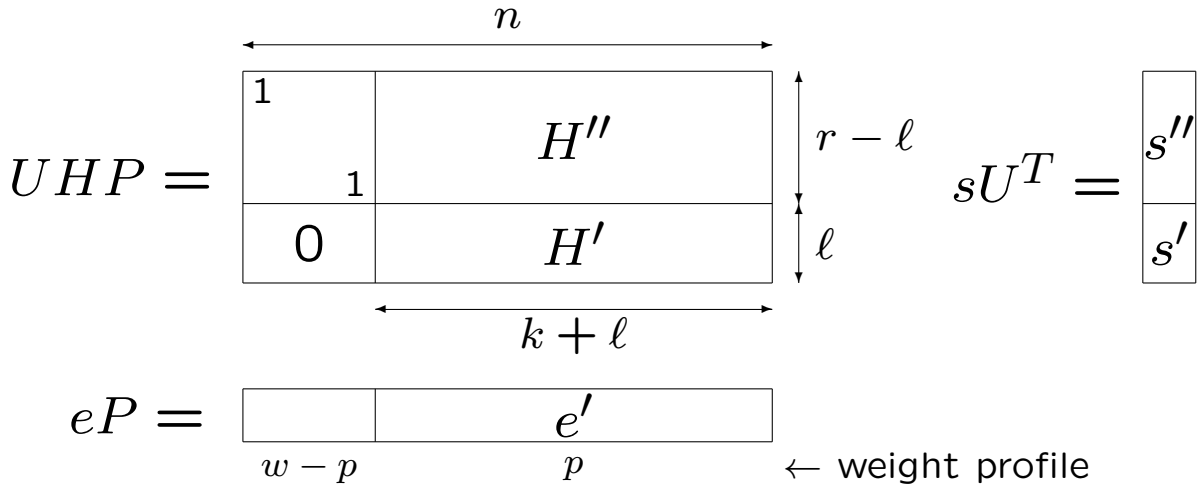
Repeat:
1. Permutation + partial Gaussian elimination
2. Find many e' such that $H'e' = s'$
3. For all good e', test $\mathrm{wt}(s'' + H''e') \leq w - p$

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 ℓ and p

Generalized Information Set Decoding – Workfactor



Assuming the Gaussian elimination cost is not significant

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

column operations up to a small constant factor. Simplifies to

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

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 & S., 09]
- Ball-collision decoding [Bernstein, Lange, & Peters, 11]
- Asymptotic exponent improved [May, Meurer, & Thomae, 11]
- Decoding one out of many [S., 11]
- Even better asymptotic exponent [Becker, Joux, May, & Meurer, 12]

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 $\rightarrow$ 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

  $\rightarrow$ 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 & S., 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

$H \in \{0,1\}^{r \times n}$ 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 $h(M) = s \in \{0,1\}^r$
- 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 $h(M) = s \in \{0,1\}^r$
- 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 144×65536 ($\approx 1.2\,\text{MB}$)

Let $s \in_R \{0,1\}^{144}$, let w be the minimal weight of e such that $s = eH^T$

- $w \leq 9$ with probability $\approx 3\,10^{-6}$ (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 $H \in \{0,1\}^{r \times n}$ (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 (λ times) longer hash of the message $h(M) = (s_1, ..., s_\lambda)$
- sign all λ syndromes $\rightarrow$ security back to $\frac{1}{2}tm$
- λ must be 3 or 4 (do not need to grow with the security parameter)

Signature length & cost and verification cost all multiplied by λ

CFS Digital Signature – Implementation

- [Landais & S., 12] Software implementation of parallel-CFS

  $(m, t) = (20, 8)$, $\lambda = 3 \rightarrow 80$ bits security

  Key size: 20 MB, one signature in ≈ 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: $H \in \{0,1\}^{r \times n}$, 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
Commitment	$\sigma \leftarrow \mathcal{S}_n$ $y \leftarrow \{0,1\}^n$	$\xrightarrow{c_0, c_1, c_2}$	
Challenge		$\xleftarrow{b}$	$b \leftarrow \{0,1,2\}$
Answer		$\xrightarrow{A_b}$	check commitments

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

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

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**)

$\rightarrow$ For a security level S, $S/\log_2(3/2) \approx 1.7S$ rounds are needed (80 bits security $\rightarrow$ 137 rounds, 128 bits security $\rightarrow$ 219 rounds)

$\rightarrow$ Can be transformed into a signature (Fiat-Shamir NIZK)

$\rightarrow$ A tight security reduction to syndrome decoding

Signing with Stern ZK Protocol

	Prover		Verifier
Commitment	$\sigma_i \leftarrow \mathcal{S}_n$ $y_i \leftarrow \{0,1\}^n$	$c_{0,i}, c_{1,i}, c_{2,i}$ $\longrightarrow$	
Challenge		$\longleftarrow$ b_i	$b_i \leftarrow \{0,1,2\}$
Answer		$A_{b_i,i}$ $\longrightarrow$	check commitments

- Draw σ_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