The Design of Code-based Cryptosystems

Indocrypt 2009 tutorial

Nicolas Sendrier

INRIA

December 13, 2009, New Delhi, India

Purpose of this Tutorial

Provide an overview of the most important aspects of code-based cryptography in order to

- understand the main code-based cryptosystems
- design new secure and efficient systems

Outline

I. Introduction to Codes and Code-based Cryptography

II. Security Reduction to Difficult Problems

III. Implementation

IV. Practical Security - The Attacks

V. Public Key – Conclusions

VI. Symmetric Code-based Cryptography – “What If We Don’t Need a Trapdoor”

\+ Some Facts about Binary Goppa Codes

Using Codes for Cryptography - Basic Idea

Error correcting codes consist in appending some redundancy to a block of data (the resulting - larger - block is called a codeword) in order to resist to transmission errors

Provided the number of errors is not too large, the process of adding random errors to a codeword

- is reversible in an information theoretic point of view
- is computationally intractable in general

This provides the basis for a cryptographic one way function

Algebraic coding theory provides encoding techniques with polynomial time error correcting procedures

This allows the introduction of trapdoors by choosing codes with a proper algebraic structure

I. Introduction to Codes and Code-based Cryptography

Notations

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

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

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

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

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

$\mathcal{B}_n(x, t) = \{y \in \mathbf{F}_q^n \mid \mathrm{dist}(x, y) \leq t\}$ the ball of center x and radius t

$\mathcal{S}_n(x, t) = \{y \in \mathbf{F}_q^n \mid \mathrm{dist}(x, y) = t\}$ the sphere of center x and radius t

$\mathcal{B}_n(\mathbf{0}, t)$ the words of weight $\leq t$

$\mathcal{S}_n(\mathbf{0}, t)$ the words of weight t

Linear Error Correcting Codes

A q-ary $\mathcal{C}(n,k)$ code 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 $\mathcal{C}$

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

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 (n,k) code and let $r = n - k$ denote its codimension

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

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

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

Inverting the syndrome: (H^* a right inverse of H)

- Given $s \in \mathbf{F}_q^r$ the word $y = sH^{*T} \in \mathbf{F}_q$ admits s as syndrome
- If $H = (\mathrm{Id} \mid R)$ is systematic then $y = (s, 0) \in S_H^{-1}(s)$
- Finding a word of $S_H^{-1}(s)$ of smallest weight (coset leader) is another matter (NP-hard)

Decoding

Let $\mathcal{C}$ be a q-ary (n, k) code of minimum distance d

Minimum distance of $\mathcal{C}$: $\text{dmin}(\mathcal{C}) = \min\{\text{wt}(x) \mid x \in \mathcal{C}, x \neq \mathbf{0}\}$

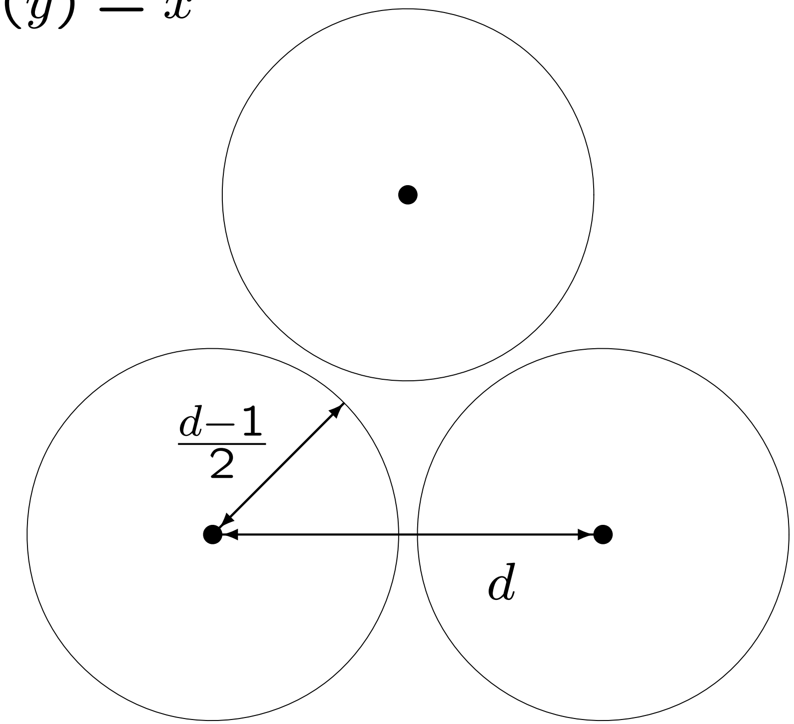
A decoder is a mapping $\Phi_{\mathcal{C}} : \mathbf{F}_q^n \to \mathcal{C}$

A decoder $\Phi_{\mathcal{C}}$ is t-bounded if for all $x \in \mathcal{C}$ and all $y \in \mathbf{F}_q^n$

$$\text{dist}(x, y) \leq t \Rightarrow \Phi_{\mathcal{C}}(y) = x$$

If a decoder is t-bounded, any element of $\mathcal{B}_n(x, t)$ with $x \in \mathcal{C}$ is decoded as x

A t-bounded decoder exists if and only if $t \leq \frac{d-1}{2}$



Syndrome Decoding

Let $\mathcal{C}$ be a q-ary (n, k) code a minimum distance d and let $H \in \mathbf{F}_q^{r \times n}$ be a parity check matrix of $\mathcal{C}$

$\Psi_H : \mathbf{F}_q^r \to \mathbf{F}_q^n$ is H-syndrome decoder if $\Psi_H(s)H^T = s$ for all $s \in \mathbf{F}_q^r$

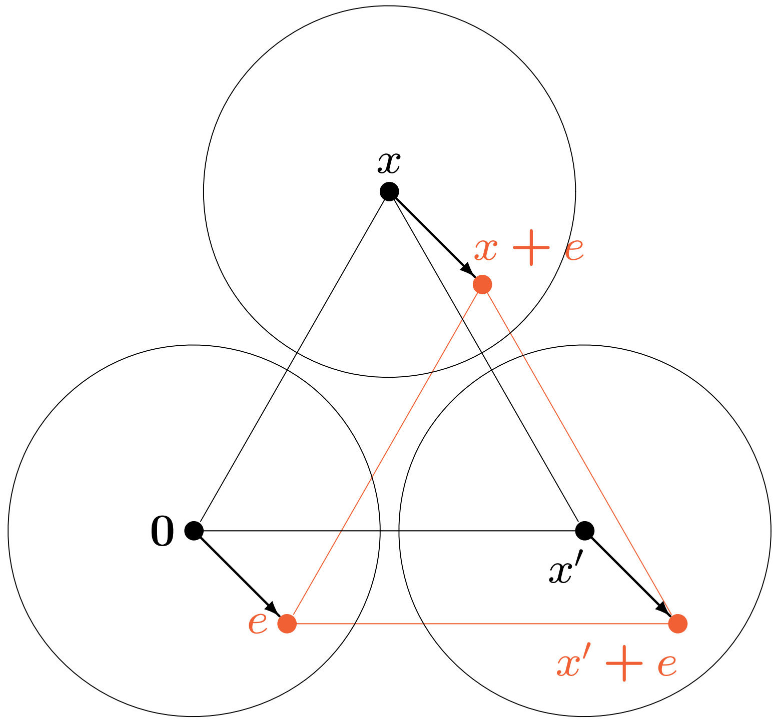
Ψ_H is t-bounded if for all $s \in \mathbf{F}_q^r$ and for all $e \in \mathbf{F}_q^n$

$$\text{wt}(e) \le t \Rightarrow \Psi_H(eH^T) = e$$

The coset elements (in color) share the same syndrome $s = eH^T$.

On input $s \in \mathbf{F}_q^r$, a t-bounded decoder returns the element of the coset $S_H^{-1}(s)$ in $\mathcal{B}_n(\mathbf{0}, t)$, if any

A t-bounded decoder exists if and only if $t \le \frac{d-1}{2}$



Decoding vs. Syndrome Decoding

Let $\mathcal{C}$ be a q-ary (n, k) code and let $H \in \mathbf{F}_q^{r\times n}$ be a parity check matrix

Ideally, a decoder looks for a codeword closest to its input while a syndrome decoder looks for a word of smallest weight in a coset.

Let Ψ_H be a syndrome decoder, we define for all $y \in \mathbf{F}_q^n$

$$\phi(y) = y - e \text{ where } e = \Psi_H(yH^T)$$

Let $\Phi_{\mathcal{C}}$ be a decoder for $\mathcal{C}$, we define for all $s \in \mathbf{F}_q^r$ (let $HH^* = \mathrm{Id}$)

$$\psi(s) = y - \Phi_{\mathcal{C}}(y) \text{ where } y = sH^{*T}$$

We have $\begin{cases} \Psi_H \text{ is } t\text{-bounded} & \Rightarrow \quad \phi \text{ is a } t\text{-bounded decoder for } \mathcal{C} \\ \Phi_{\mathcal{C}} \text{ is } t\text{-bounded} & \Rightarrow \quad \psi \text{ is a } t\text{-bounded syndrome decoder} \end{cases}$

$\longrightarrow$ decoding and syndrome decoding are essentially the same thing

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 $\Phi : \{0,1\}^n \rightarrow \mathcal{C}$ for $\mathcal{C}$ $\forall y \in \{0,1\}^n, \forall x \in \mathcal{C}, (d_H(x,y) \leq t) \Rightarrow (\Phi(y) = x)$
Encryption:	$\left[\begin{array}{llcl} E_G : & \{0,1\}^k & \rightarrow & \{0,1\}^n \\ & x & \mapsto & xG + e \end{array} \right]$ with e random of weight t
Decryption:	$\left[\begin{array}{llcl} D_\Phi : & \{0,1\}^n & \rightarrow & \{0,1\}^k \\ & y & \mapsto & \Phi(y)G^* \end{array} \right]$ 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 $\Psi : \{0,1\}^r \rightarrow \{0,1\}^n$ $\forall e \in \{0,1\}^n, (\mathrm{wt}(e) \leq t) \Rightarrow (\Psi(eH^T) = e)$
Encryption:	$\left[\begin{array}{llll} E_H: & \mathcal{S}_n(\mathbf{0},t) & \rightarrow & \{0,1\}^r \\ & e & \mapsto & eH^T \end{array}\right]$
Decryption:	$\left[\begin{array}{llll} D_\Psi: & \{0,1\}^r & \rightarrow & \mathcal{S}_n(\mathbf{0},t) \\ & s & \mapsto & \Psi(s) \end{array}\right]$ s must be a cryptogram

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

McEliece/Niederreiter Security

We must make sure that the following two problems are difficult enough to an attacker:

1. Retrieve a t-bounded decoder from the public key, a generator matrix or a parity check matrix

   The legal user must be able to decode so the algebraic structure exists, it must remain hidden to the cryptanalyst

2. Decode t errors in a random binary (n, k) code

   Without the algebraic structure, the cryptanalyst can only use generic technique to decode

   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 2008 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. 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 $\text{wt}(e) \le w$ 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 $\text{wt}(e) \le w$ 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 $\text{wt}(e) \le \dfrac{r}{\log_2 n}$ and $eH^T = s$

Open problem: average case complexity (Conjectured difficult)

Hard Structural Problems

Goppa code Distinguishing NP

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

Question: Is $\left\{x \in \{0,1\}^n \mid xH^T = 0\right\}$ a binary Goppa code?

Goppa code Reconstruction

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

Output: (L,g) such that $\Gamma(L,g) = \left\{x \in \{0,1\}^n \mid xH^T = 0\right\}$

- NP: the property is easy to check given (L,g)
- Completeness status is unknown
- Tightness: gap between decisional and computational problems

Goppa Code Distinguisher

For given parameters n, k

Let $\mathcal{G}$ denote the set of all generator matrices of a Goppa code.
For any program $\mathcal{D} : \{0,1\}^{k\times n} \rightarrow \{\mathsf{true}, \mathsf{false}\}$, we define the event

$$\mathcal{T}_{\mathcal{D}} = \{G \in \Omega \mid \mathcal{D}(G) = \mathsf{true}\}$$

in the sample space $\Omega = \{0,1\}^{k\times n}$ uniformly distributed

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

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

Decoding Adversary

For given parameters n, k and t

For any program $\mathcal{A} : \{0,1\}^n \times \{0,1\}^{k\times n} \to \{0,1\}^k$, we define the event

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

in the sample space $\Omega = \{0,1\}^k \times \{0,1\}^{k\times n} \times \mathcal{S}_n(\mathbf{0}, t)$ uniformly distributed

$\mathcal{A}$ is a (T, ε)-decoder if

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

Adversary against McEliece

For given parameters n, k and t

Let $\mathcal{G}$ denote the set of all generator matrices of a Goppa code.
For any program $\mathcal{A} : \{0,1\}^n \times \{0,1\}^{k\times n} \rightarrow \{0,1\}^k$, we define the event

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

in the sample space $\Omega = \{0,1\}^k \times \{0,1\}^{k\times n} \times \mathcal{S}_n(\mathbf{0}, t)$ uniformly distributed

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

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

If there exists a (T, ε)-adversary then there exists either

- a $(T, \varepsilon/2)$-decoder,
- or a $(T + O(n^2), \varepsilon/2)$-distinguisher,

Adversary against Niederreiter

For given parameters n, r and t

Same thing with a slightly different adversary
$\mathcal{A} : \{0,1\}^r \times \{0,1\}^{r\times n} \rightarrow \mathcal{S}_n(\mathbf{0},t)$, with $\Omega = \mathcal{S}_n(\mathbf{0},t) \times \{0,1\}^{r\times n}$ and

$$\mathcal{S}_{\mathcal{A}} = \{(e,H) \in \Omega \mid \mathcal{A}(eH^T, H) = e\}$$

$\mathcal{A}$ is a (T,ε)-decoder if

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

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

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

If there exists a (T,ε)-adversary then there exists either

- a $(T,\varepsilon/2)$-decoder,
- or a $(T+O(n^2),\varepsilon/2)$-distinguisher,

One Way Encryption Schemes

A scheme is OWE (One Way Encryption) if the all attacks are intractable in 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 un unknown message x (*i.e.* $\quad y = xG + e$)

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 in 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.

III. Implementation

A Remark on Niederreiter Encryption Scheme

In Niederreiter's system the encryption procedure is:

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

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

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

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

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

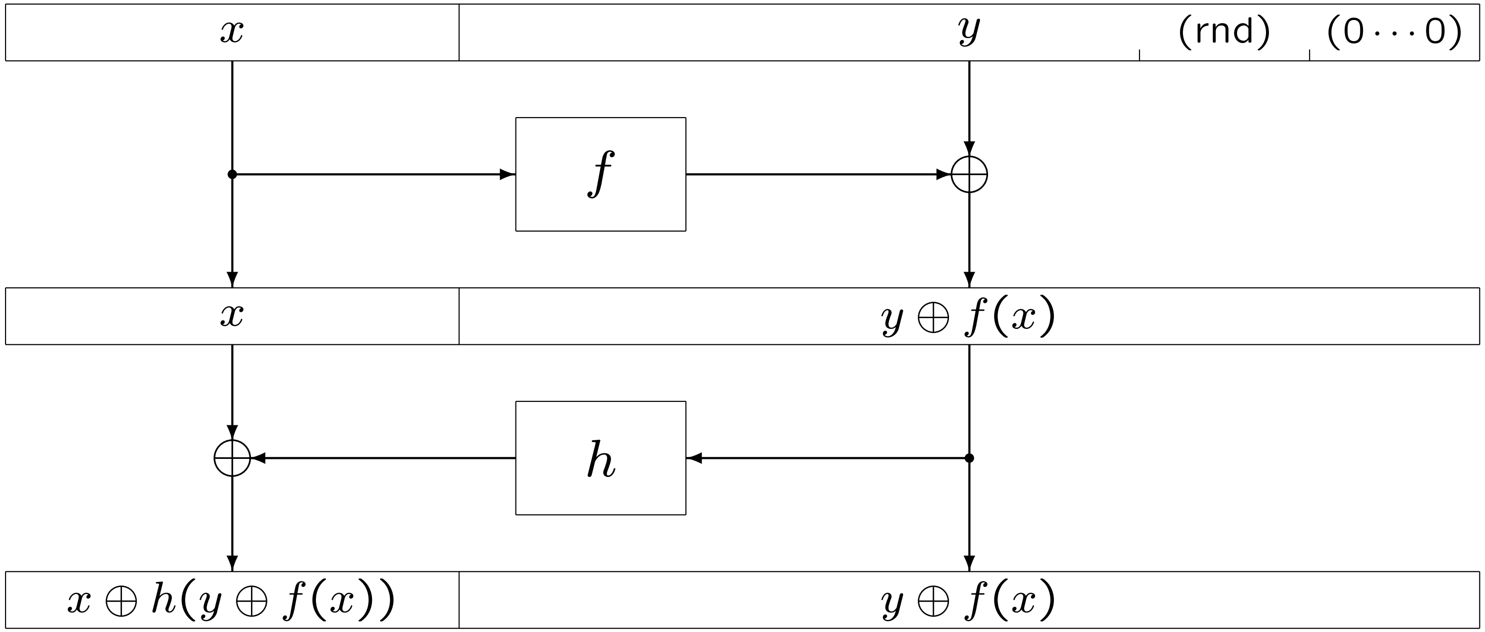
  The system remains OWE, puzzling but true!

  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 $\left\{\begin{array}{l} a = x \oplus h(y \oplus f(x)) \\ b = y \oplus f(x) \end{array}\right. \Leftrightarrow \left\{\begin{array}{l} x = a \oplus h(b) \\ y = b \oplus f(a \oplus h(b)) \end{array}\right.$

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

Some Set of Parameters

m, t	McEliece		Niederreiter		Hybrid		key	secur.
	cipher	clear	cipher	clear	cipher	clear	size	bits*
$10, 50$	1024	524	500	284	1024	808	32 kB	60
$11, 32$	2048	1696	352	233	2048	1929	73 kB	86
$12, 21$	4096	3844	252	185	4096	4029	118 kB	87
$12, 40$	4096	3616	480	320	4096	3936	212 kB	127
$13, 18$	8192	7958	234	180	8192	8138	227 kB	91
$13, 29$	8192	7815	377	273	8192	8088	360 kB	128

* logarithm in base 2 of the cost of the best known attack

key size is given for a key in systematic form

HyMES – Encryption/Decryption Speed

m, t	cycles/byte encrypt	cycles/byte decrypt	key size	security
$10, 50$	243	7938	32 kB	60
$11, 32$	178	1848	73 kB	86
$12, 21$	126	573	118 kB	87
$12, 41$	164	1412	212 kB	130
$13, 18$	119	312	227 kB	91
$13, 29$	149	535	360 kB	129
$14, 15$	132	229	415 kB	91
$15, 13$	132	186	775 kB	90
$16, 12$	132	166	1532 kB	91

AES: 15-20 cycles/byte

RSA 2048: 834 for encryption, 55922 for decryption

(All timings on Intel Core 2 processor)

IV. 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

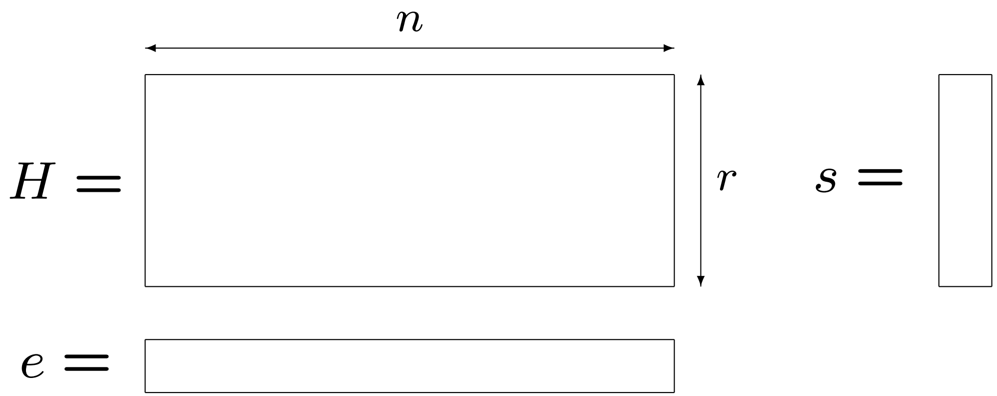
Key attacks. Most proposals using families other than binary Goppa codes have been broken

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

Syndrome Decoding – Problem Statement

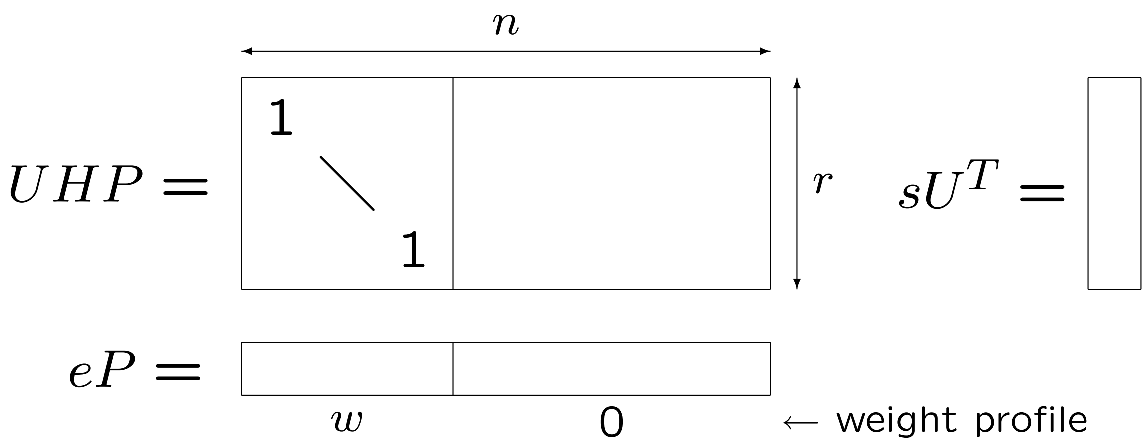
Computational Syndrome Decoding

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



Typically $w \ll r < n$ and we wish to find a few (w) columns of H which add to some given s.

Information Set Decoding



Repeat:

- Pick a permutation matrix P and compute U to obtain the above systematic form
- If $\mathrm{wt}(sU^T) \leq w$ then $e = (sU^T, 0)P^T$ is a solution ($\rightarrow$ exit)

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

Total cost: $\approx r^2 n (n/r)^w$

Information Set Decoding – Generalized

A long story:

- Relax the weight profile: [Lee & Brickell, 88]
- Compute sums on partial columns first: [Leon, 88]
- Use the birthday attack: [Stern, 89]
- First "real" implementation: [Canteaut & Chabaud, 98]
- Initial McEliece parameters broken: [Bernstein, Lange & Peters, 08]
- Asymptotic bounds: [Bernstein *et al.*, 09]
- Lower bounds: [Finiasz & S., 09]

$$UHP = \overbrace{\left[\begin{array}{c|c} \begin{matrix}1&&\\&\ddots&\\&&1\end{matrix} & H'' \\ \hline 0 & \underbrace{H'}_{k+\ell} \end{array}\right]}^{n} \begin{array}{l} \updownarrow r-\ell \\ \updownarrow \ell \end{array} \qquad sU^T = \left[\begin{array}{c} s'' \\ \hline s' \end{array}\right]$$

$$eP = \left[\begin{array}{c|c} & e' \end{array}\right]$$

$$\begin{array}{ccc} w-p & p & \leftarrow \text{weight profile} \end{array}$$

Information Set Decoding – Generalized

$$
\begin{array}{c}
\overset{n}{\longleftrightarrow} \\
UHP = \left[\begin{array}{c|c} \begin{smallmatrix}1 & & \\ & \ddots & \\ & & 1\end{smallmatrix} & H'' \\ \hline 0 & H' \end{array}\right]
\begin{array}{l} \updownarrow r-\ell \\ \updownarrow \ell \end{array}
\qquad sU^T = \left[\begin{array}{c} s'' \\ \hline s' \end{array}\right] \\
\underset{k+\ell}{\longleftrightarrow}
\end{array}
$$

$$
eP = \left[\begin{array}{c|c} \quad\quad & e' \end{array}\right] \quad \begin{array}{c} \\ \underset{w-p}{} \quad \underset{p}{} \end{array}
$$

$w-p$ $\quad$ p $\quad$ $\leftarrow$ weight profile

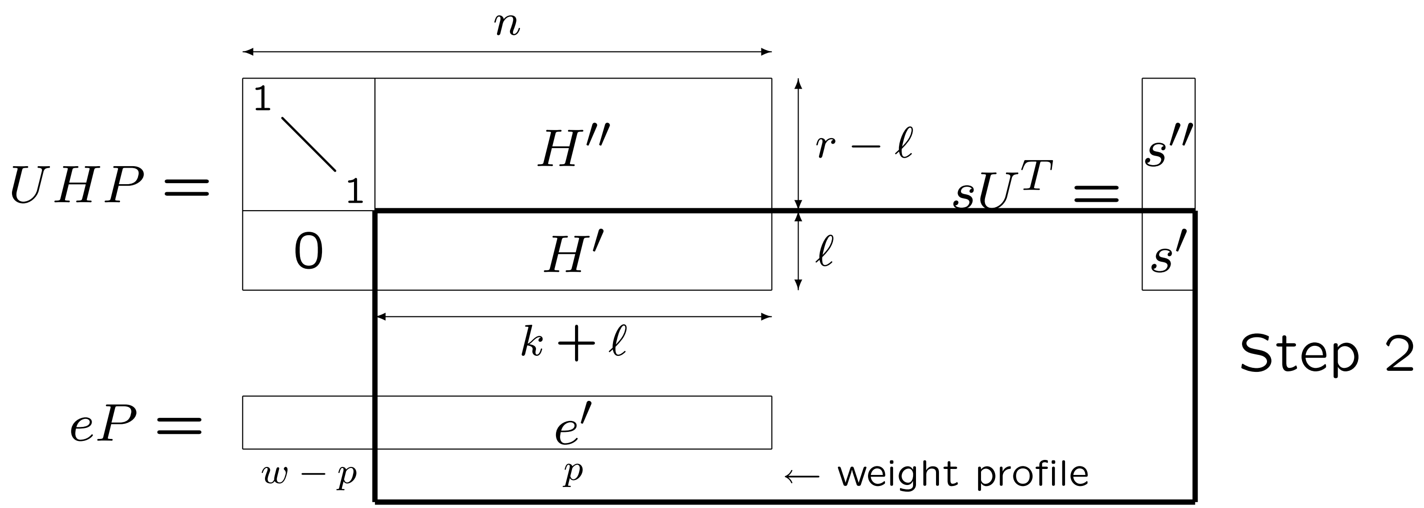
Repeat: $\left\{ \begin{array}{l} \text{1. Pick } P \text{ and compute } U \text{ to obtain the systematic form} \\ \text{2. Find many solution of weight } p \text{ to } e'H'^T = s' \\ \text{3. For all the above } e'\text{, test wt}(s'' + e'H''^T) \leq w - p \end{array} \right.$

Success probability: $\binom{r-\ell}{w}\binom{k+\ell}{p}/\binom{n}{w}$

Step 2. is performed by a birthday attack

Total cost is minimized over ℓ and p

Information Set Decoding – Generalized



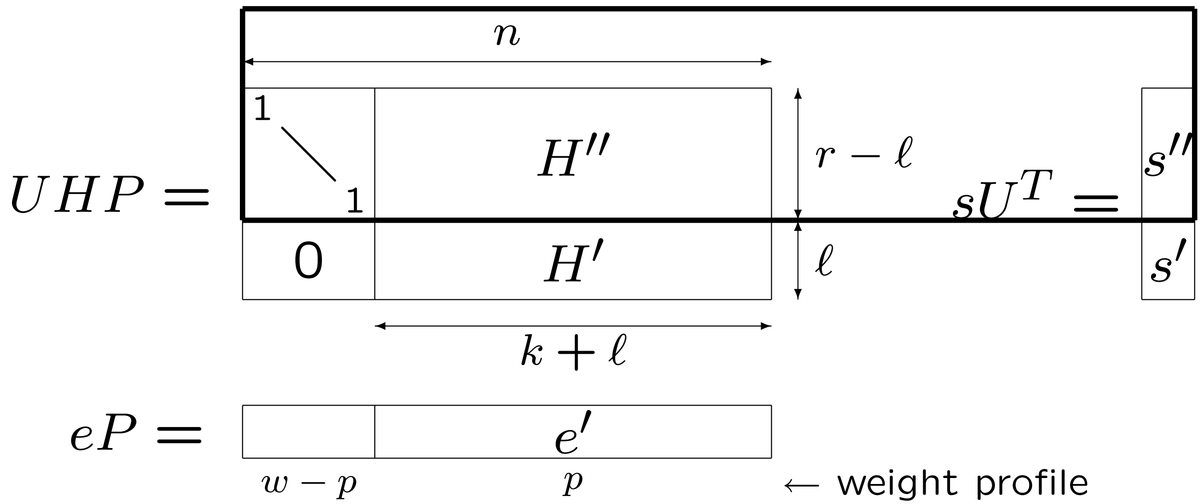
Repeat: $\begin{cases} \text{1. Pick } P \text{ and compute } U \text{ to obtain the systematic form} \\ \text{2. Find many solution of weight } p \text{ to } e'H'^T = s' \\ \text{3. For all the above } e'\text{, test wt}(s'' + e'H''^T) \leq w - p \end{cases}$

Success probability: $\binom{r-\ell}{w}\binom{k+\ell}{p}/\binom{n}{w}$

Step 2. is performed by a birthday attack

Total cost is minimized over ℓ and p

Information Set Decoding – Generalized



Step 3

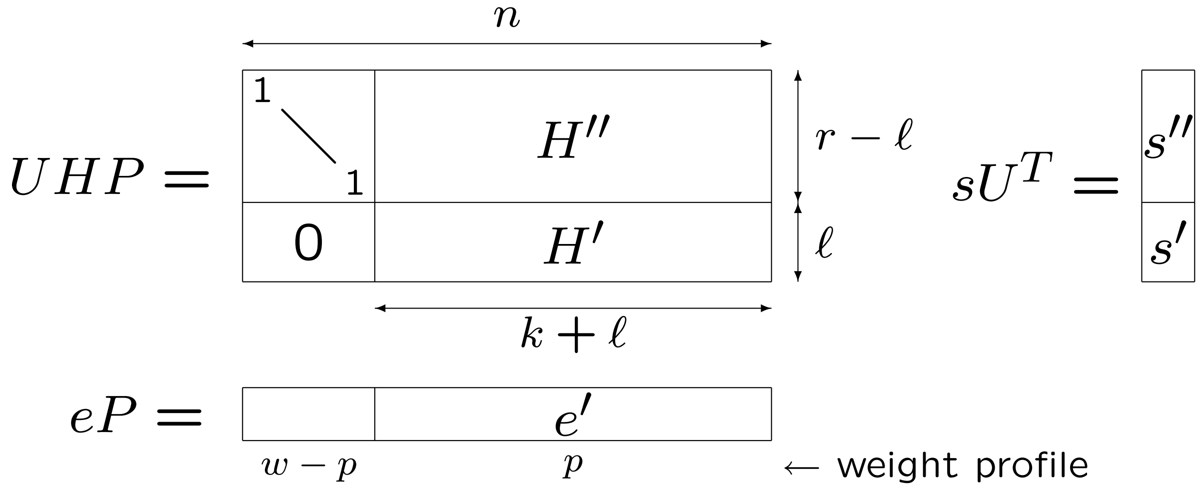
Repeat:
1. Pick P and compute U to obtain the systematic form
2. Find many solution of weight p to $e'H'^T = s'$
3. For all the above e', test wt$(s'' + e'H''^T) \leq w - p$

Success probability: $\binom{r-\ell}{w}\binom{k+\ell}{p}/\binom{n}{w}$

Step 2. is performed by a birthday attack

Total cost is minimized over ℓ and p

Information Set Decoding – Work Factor



Assuming the Gaussian elimination is free

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

where K_{w-p} is the cost for checking $\mathrm{wt}(s'' + e'H''^T) \leq w - p$. The value of ℓ minimizing the formula can be computed and we have

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

Key Security

Finding families of codes whose structure cannot be recognized is 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)

Attacks on Goppa Codes

The only known attacks on binary Goppa codes are exhaustive. Let $\Gamma(L, g)$ be the secret code.

- [Gibson, 91] Enumerate all possible supports and compute the generator polynomial by using $g(z)|\sigma'_a(z)$ for all codeword a
- [Loidreau & S., 01] Enumerate the generators (irreducible polynomials of degree t) build a generator of the the corresponding Goppa code with any support and test the equivalence with the support splitting algorithm [S., 00]

Message Security vs. Key Security

The following table shows the huge gap between the best decoding attack and best key attack

m, t	sizes					security (in bits)	
	McEliece		Niederreiter		public key (syst.)		
	cipher	clear	cipher	clear		mess.	key
$10, 50$	1024	524	500	284	32 kB	60	491
$11, 32$	2048	1696	352	233	73 kB	86	344
$12, 40$	4096	3616	480	320	212 kB	127	471

Can we trade some of the extra key security for a smaller key size?

Key Size Reduction

Attempts for shorter public key

- Rank metric [Gabidulin *et al.*, 91]

  Lot of contributions, finally seriously weakened by [Overbeck ,07]

  Still breathing ?

- Codes with structure
  - Quasi-cyclic codes [Gaborit, 05], broken by [Otmani *et al.*, 08]
  - Quasi-cyclic codes [Berger *et al.*, 09] then Quasi-dyadic Goppa codes [Misoczki & Barreto, 09]
    weakened by [Otmani *et al.*], unpublished (using Gröbner basis)

Hard open problem, but through the recent works it seems possible to understand the weaknesses of the existing approaches. . .

V. Public Key – Conclusions

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, Cayrel & Gaborit, 08],

Conclusion for Public Key Code-based Cryptosystems

- Good security reduction

  partly heuristic though:
  - nothing proven on the average case complexity of decoding
  - indistinguishability of Goppa codes needs more investigations

- The best attacks are decoding attacks

- Attacks on the public key need more attention

- Open problems: mainly related to the key
  - find families of secret codes other than Goppa
  - find secure ways to reduce the key size (structured codes)

VI. Symmetric Code-based Cryptography – “What If We Don’t Need a Trapdoor”

Average Case Complexity of Decoding

Decoding in random linear code is an old algorithmic problem from coding theory. It is known to be hard in the worst case (NP-complete).

Though this is not assessed by any theoretical result, it is believed to be hard in the average case. Coding theorists have tried very hard, for several decades, to produce efficient generic decoders and have only found algorithms with an exponential cost on almost all instance.

Improving those algorithms even with limitations would have a strong impact in the theory but may be also in the practice of error correcting codes. It is thus relatively safe to assume the average case difficulty of decoding.

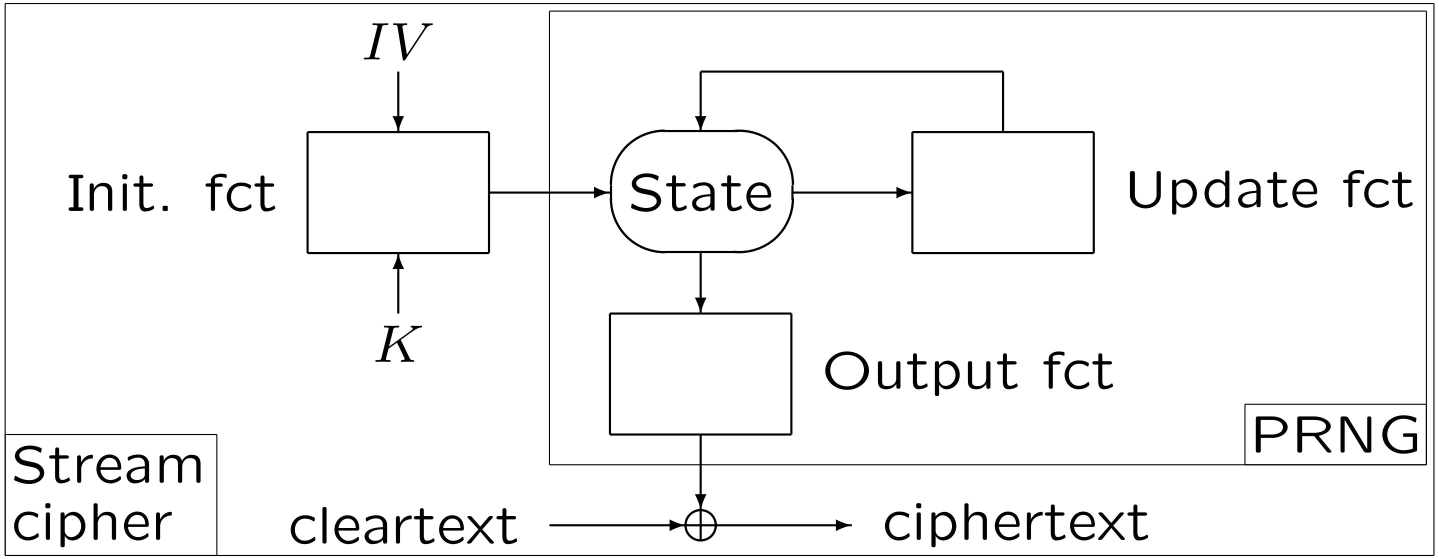
The syndrome mapping $e \mapsto eH^T$, when e has a small weight and H is chosen randomly, provides a very efficient one way function whose security is reduced to the above assumption.

Symmetric Code-based Cryptosystems

We will present two applications of code-based one-way functions

- Pseudo-Random Number Generators (PRNG) and stream ciphers
- Cryptographic hash functions

PRNG with Codes

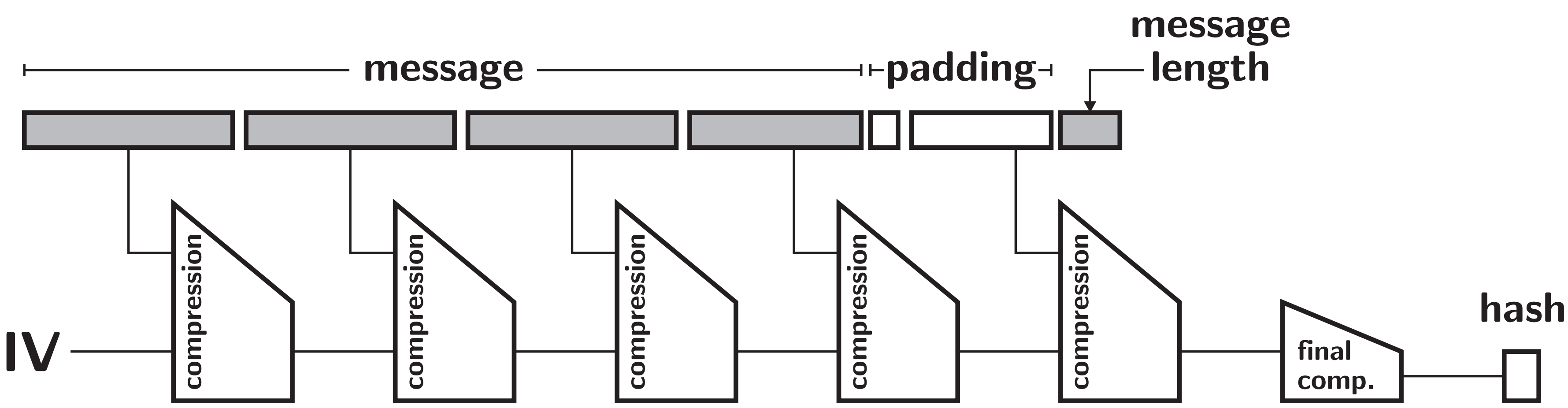


A stream cipher can be build from a PRNG but adding an initialization and xoring the keystream (output of the PRNG) to the cleartext

[Fischer & Stern, 96] propose a PRNG where the update function is a syndrome mapping with a few bits of output at each update

[Gaborit, Lauradoux & S., 07] use a syndrome mapping also for the output as well as an initialization function (also syndrome-based). In addition, all the matrix used are quasi-cyclic.

FSB: Fast Syndrome-Based Hash Function



The compression function uses a syndrome mapping

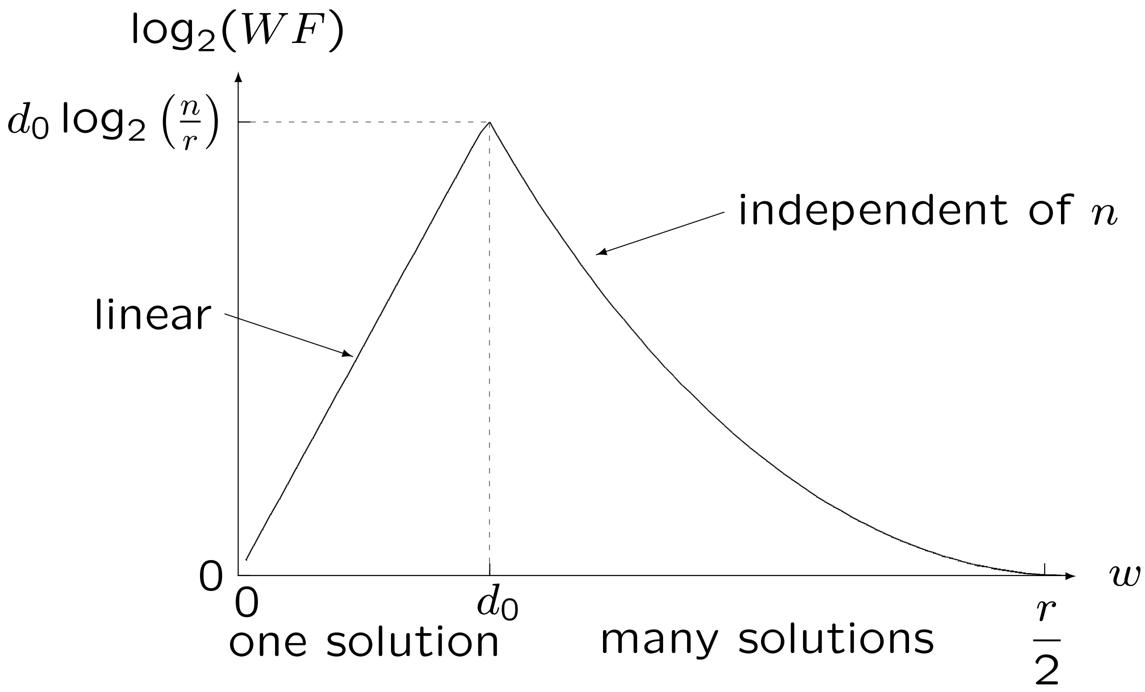
$$\begin{array}{ccccc} f: \{0,1\}^\ell & \rightarrow & \mathcal{S}_n(\mathbf{0}, w) & \rightarrow & \{0,1\}^r \\ x & \mapsto & \varphi(x) = e & \mapsto & eH^T \end{array}$$

In order to achieve compression, the error weight must be much higher than in all other code-based systems (f is surjective)

Difficult: we are not any more in a usual decoding problem. We must check that syndrome decoding remains hard

Information Set Decoding for Larger Weight

This plot describes the evolution of the cost (log) of ISD for a fixed code (of length n and codimension r) when the weight increases



The maximum is reached for the Gilbert-Varshamov distance, that is when $2^r \approx \binom{n}{d_0}$. Since we must have $\binom{n}{w} > 2^r$ to achieve compression, we must have $w > d_0$.

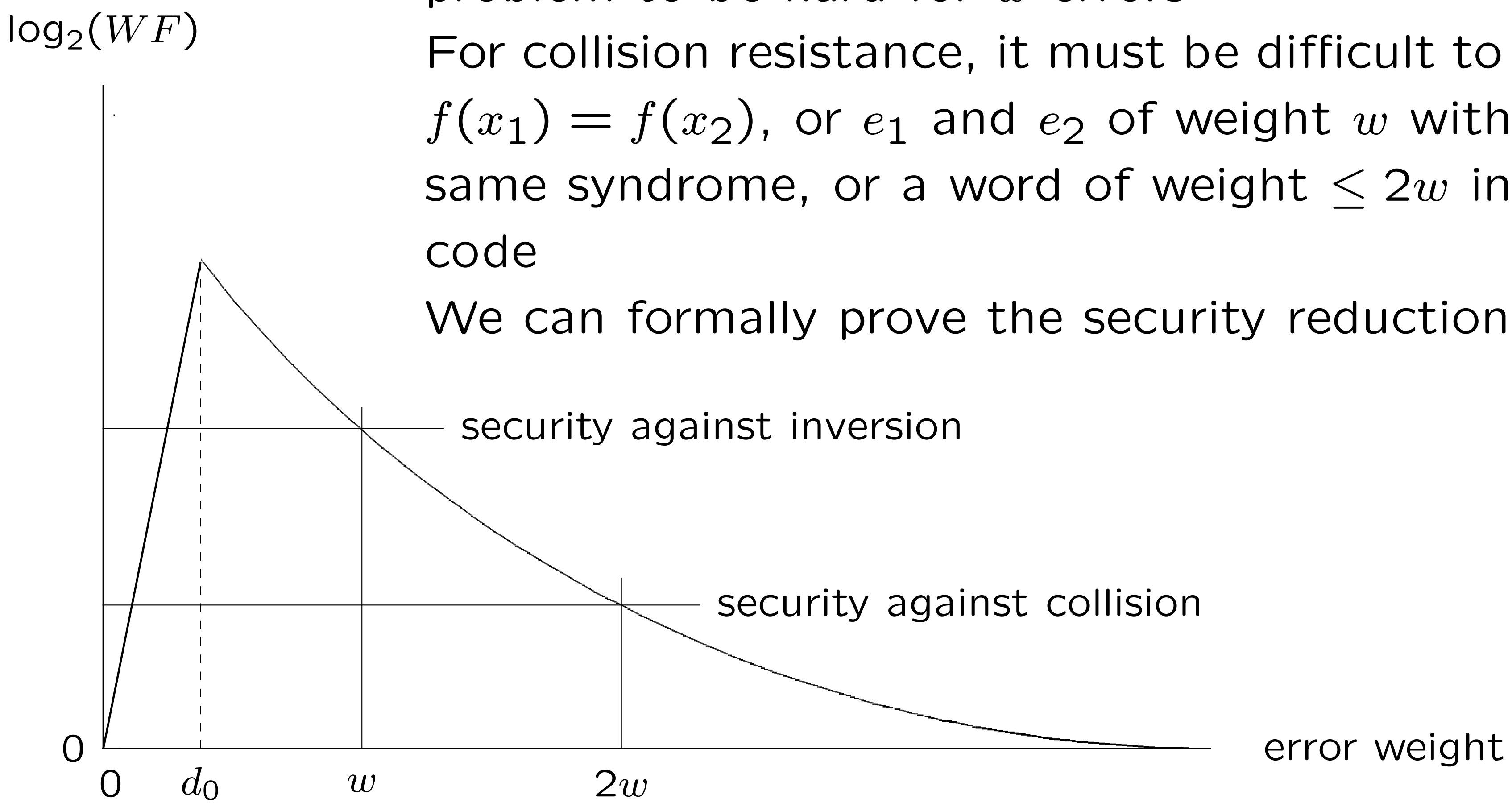
Parameter selection for the FSB hash function

$$\begin{array}{rcccc} f: & \{0,1\}^\ell & \rightarrow & \mathcal{S}_n(\mathbf{0},w) & \rightarrow & \{0,1\}^r \\ & x & \mapsto & \varphi(x)=e & \mapsto & eH^T \end{array}$$

For inversion resistance, we need the decoding problem to be hard for w errors

For collision resistance, it must be difficult to find $f(x_1)=f(x_2)$, or e_1 and e_2 of weight w with the same syndrome, or a word of weight $\leq 2w$ in the code

We can formally prove the security reduction



FSB – Conclusions

When we reach high values of w, another attack has to be considered: the Generalized Birthday Algorithm (GBA) [Wagner, 02]

It does not change the security reduction, but the parameter selection process must take GBA into account.

Note also that the constant weight word mapping φ encodes only regular words (for speed) and the codes we use are quasi-cyclic. The impact on security is acceptable.

The function was submitted to the SHA-3 competition but did not reach the second round. It was not broken but was 10 to 20 times slower than ad-hoc designs.

Thank you for your attention