Code-based One Way Functions

Nicolas Sendrier

INRIA Rocquencourt, projet CODES

Ecrypt Summer School
Emerging Topics in Cryptographic Design and Cryptanalysis

I. Introduction

Binary linear code

$\mathcal{C}$ a (n,k) binary linear code $\quad \begin{cases} n & \text{the length} \\ k & \text{the dimension} \\ r = n-k & \text{the co-dimension} \end{cases}$

Generator matrix G (size $k \times n$) $\quad \mathcal{C} = \{uG \mid u \in \{0,1\}^k\}$

Parity check matrix H (size $r \times n$) $\quad \mathcal{C} = \{x \in \{0,1\}^n \mid xH^T = 0\}$

—

For any $y \in \{0,1\}^n$, yH^T is the syndrome of y relatively to H

The set $y + \mathcal{C} = \{y + x \mid x \in \mathcal{C}\}$ is a coset of $\mathcal{C}$

We have $y + \mathcal{C} = \{v \in \{0,1\}^n \mid vH^T = yH^T\}$

Decoding a linear code

Decoding: given $y \in \{0,1\}^n$, find a codeword $x \in \mathcal{C}$ closest to y (for the Hamming distance)

Find $e \in \{0,1\}^n$ of minimal Hamming weight such that (equivalently)

(i) $x = y - e \in \mathcal{C}$

(ii) $e \in y + \mathcal{C}$

(iii) $eH^T = yH^T$

Decoding is difficult in general

The syndrome decoding problem

Berlekamp, McEliece, van Tilborg, 1978

Problem: Syndrome Decoding — NP-complete

Instance: An $r \times n$ binary matrix H, a word s of $\{0,1\}^r$ and an integer $t > 0$.

Question: Is there a word of weight $\leq t$ in $\{e \mid eH^T = s\}$?

Easy for small (constant) or for large values of t (i.e. $t \gtrsim r/2$)

Average case complexity: no reduction is known.

Decades of research indicate that it is hard in practice.

Heuristic: most difficult if $\binom{n}{t} \approx 2^r$ (Gilbert-Varshamov bound)
(see the *Handbook of Coding Theory*, chapter 7 "Complexity issues in coding theory", by A. Barg)

Bounded decoding

What about smaller values of the error weight ?

Finiasz, 2004

Problem: GOPPA BOUNDED DECODING NP-complete
Instance: An $r \times n$ binary matrix H, a word s of $\{0,1\}^r$.
Question: Is there a word of weight $\leq \frac{r}{\log_2 n}$ in $\{e \mid eH^T = s\}$?

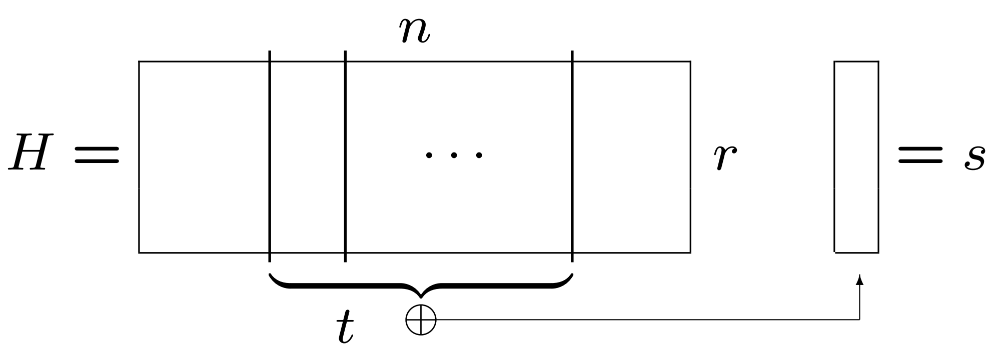
The number of errors you can decode in a binary Goppa code of length n and codimension r is $\approx r/\log_2 n$. Probably still NP-complete for $w = cr/\log_2 n$, $c > 0$.

Also considered difficult in practice in the average case.

II. Code-based one-way functions

The syndrome mapping – A simple and fast primitive

Let H be a binary $r \times n$ matrix



n a few thousand
r several hundreds
t a few dozens

Complexity: t column additions for one column of output

Let $W_{n,t}$ denote the set of words of length n and weight t. The syndrome mapping is defined as

$$\begin{array}{rrcl} S: & W_{n,t} & \longrightarrow & \{0,1\}^r \\ & e & \longmapsto & eH^T \end{array}$$

Code-based one way functions

$\mathcal{C}$ a linear code, H a parity check matrix

$$\begin{array}{rrcl} \Phi : & \mathcal{E} \times \mathcal{C} & \longrightarrow & \{0,1\}^n \\ & (e, x) & \longmapsto & x + e \end{array} \qquad \Bigg| \qquad \begin{array}{rrcl} S : & \mathcal{E} & \longrightarrow & \{0,1\}^r \\ & e & \longmapsto & eH^T \end{array}$$

Φ and S are equally difficult to invert

1) $\Phi^{-1}(y) = \left(S^{-1}(yH^T), y - S^{-1}(yH^T)\right)$

2) Let $H_0 = U \cdot H = (I \mid X)$

For any $s \in \{0,1\}^r$, the word $y = (sU^T \mid 0, \ldots, 0)$ verifies $yH^T = s$

Thus $\Phi^{-1}(y) = \left(S^{-1}(s), y - S^{-1}(s)\right)$

The error set

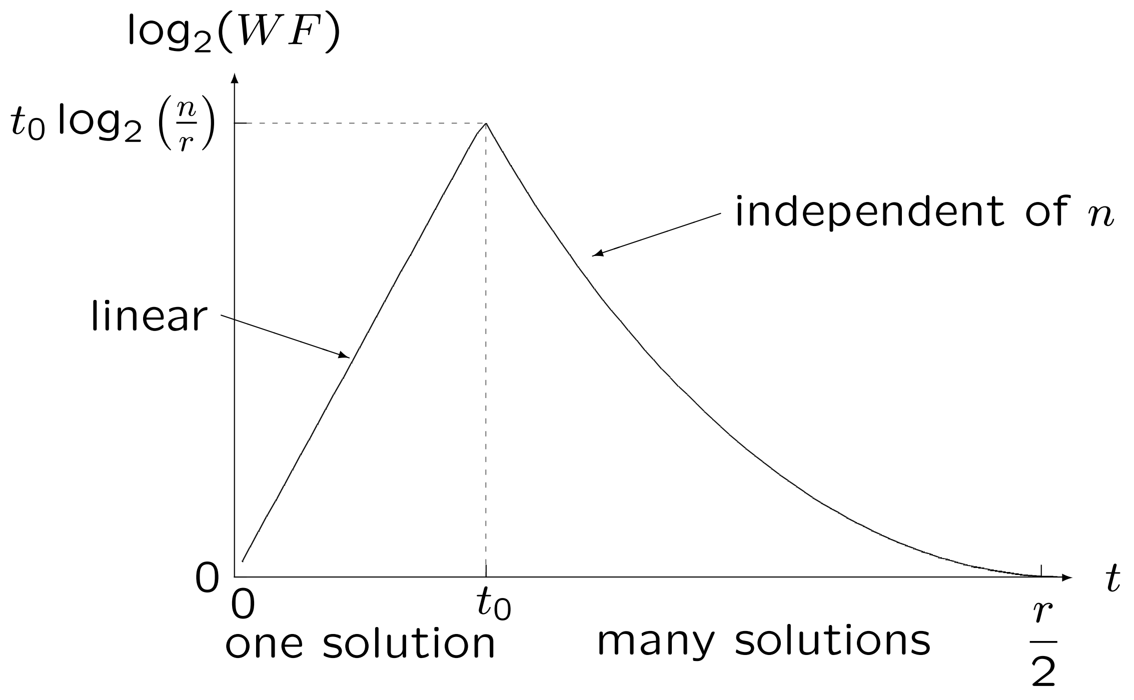
$$\begin{array}{rccl} S: & \mathcal{E} & \longrightarrow & \{0,1\}^r \\ & e & \longmapsto & eH^T \end{array}$$

Usually $\mathcal{E} = W_{n,t}$ (or $\mathcal{E} \subset W_{n,t}$) for some error weight t

- S is injective if $t \le (d-1)/2$ (d the minimum distance)
- S is surjective if $t \ge \rho$ (ρ the covering radius)
- S is (almost) never bijective

II.1 Security

Decoding attack – Information set decoding



Cost for solving $s = eH^T$ for a given H and s, with e of weigth t by information set decoding.

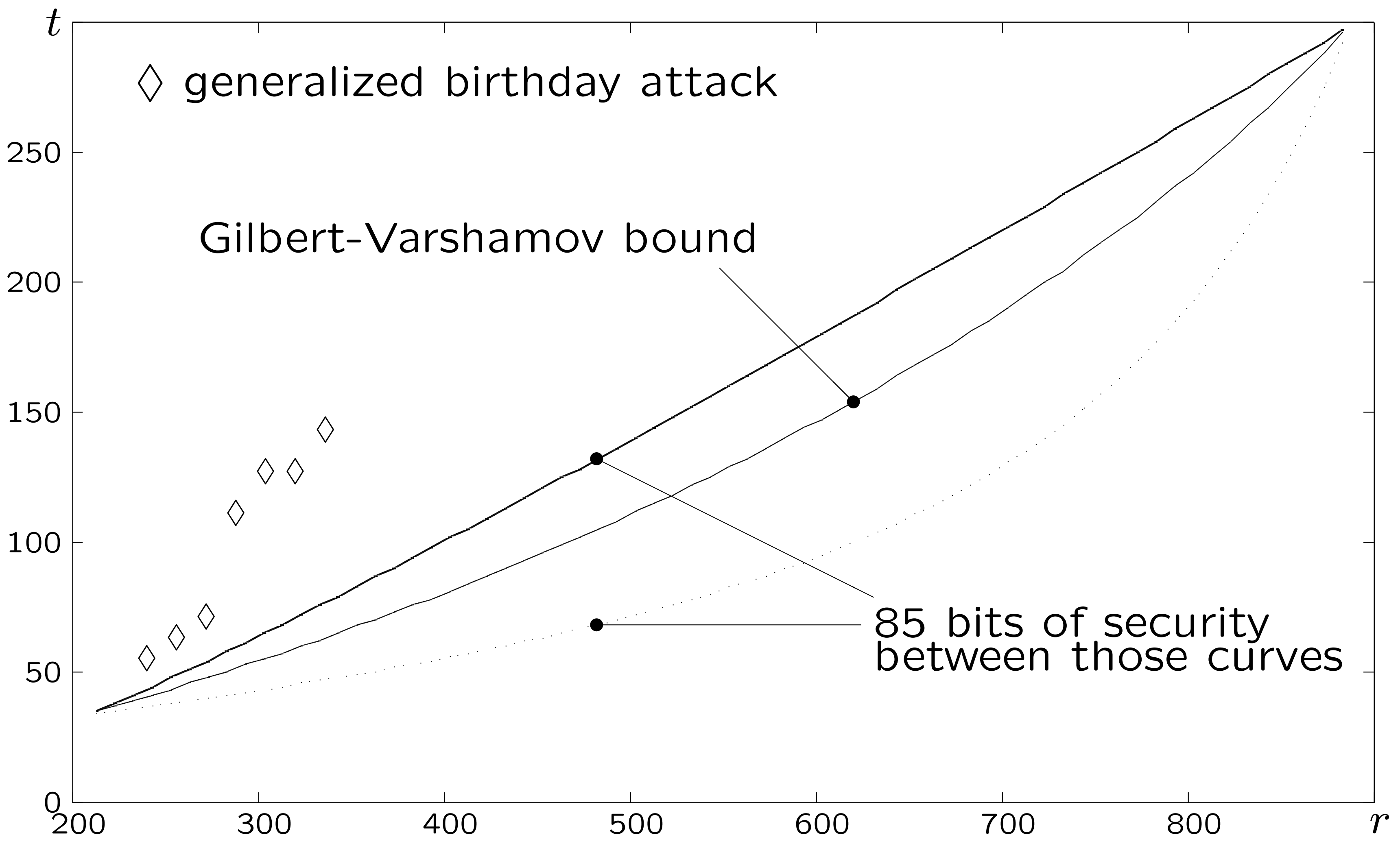
Both n and r are fixed

t_0 is such that $\binom{n}{t_0} \approx 2^r$.

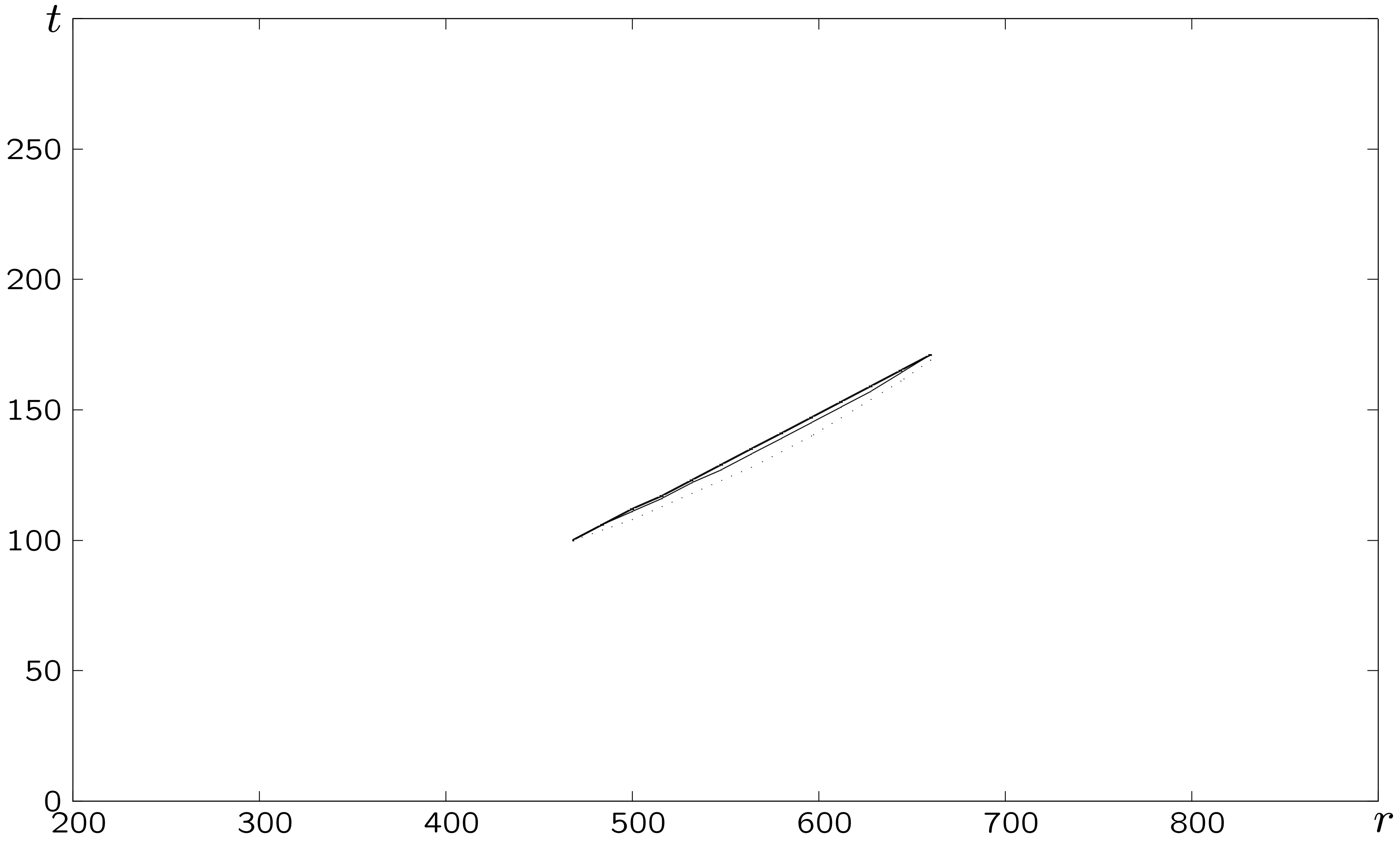
Best implementation by Canteaut and Chabaud (1998).

Information set decoding attack is the best attack when $t \leq t_0$. If $t > t_0$ the generalized birthday attack (Wagner, 2002) is sometimes better.

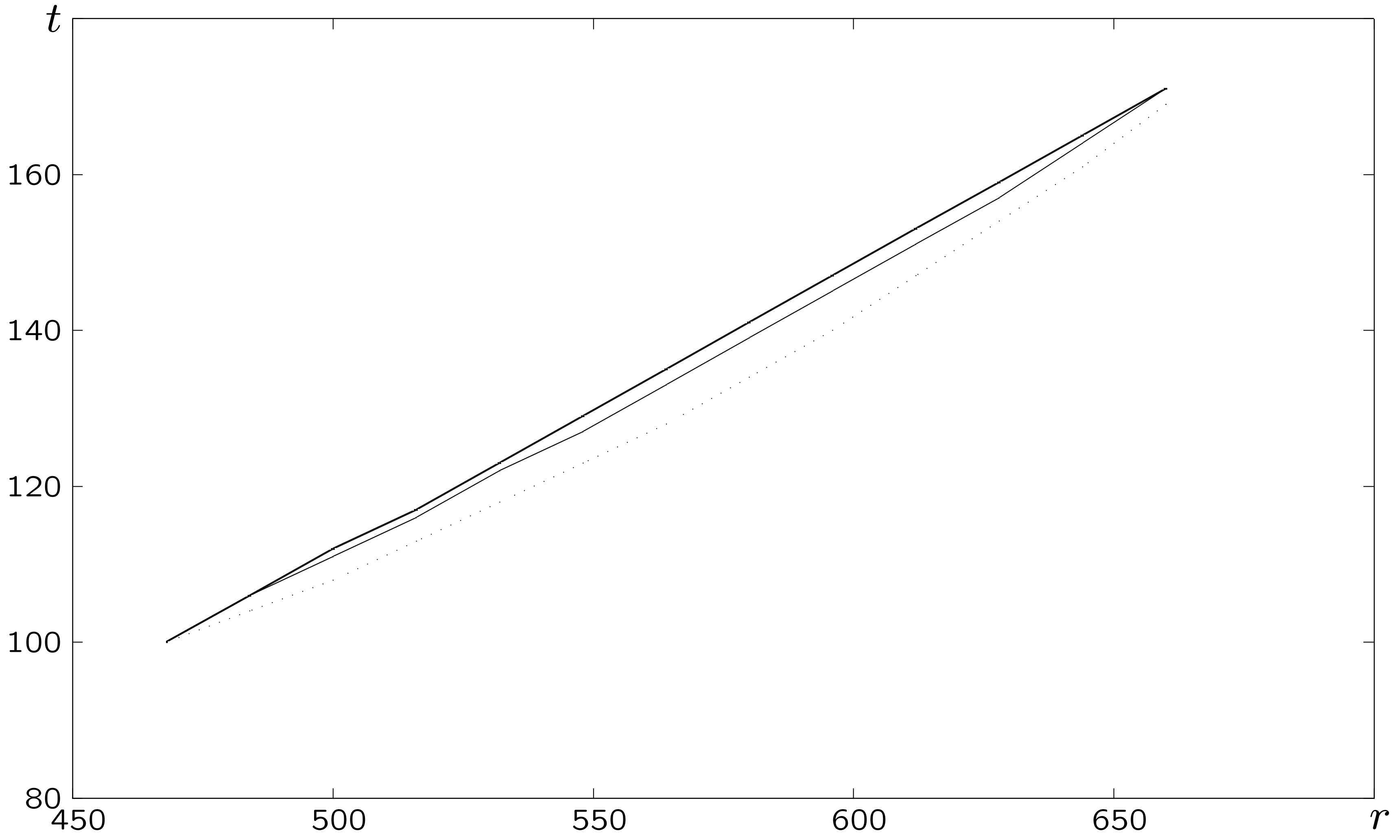
Decoding attack for $n = 1024$ and security 2^{85}



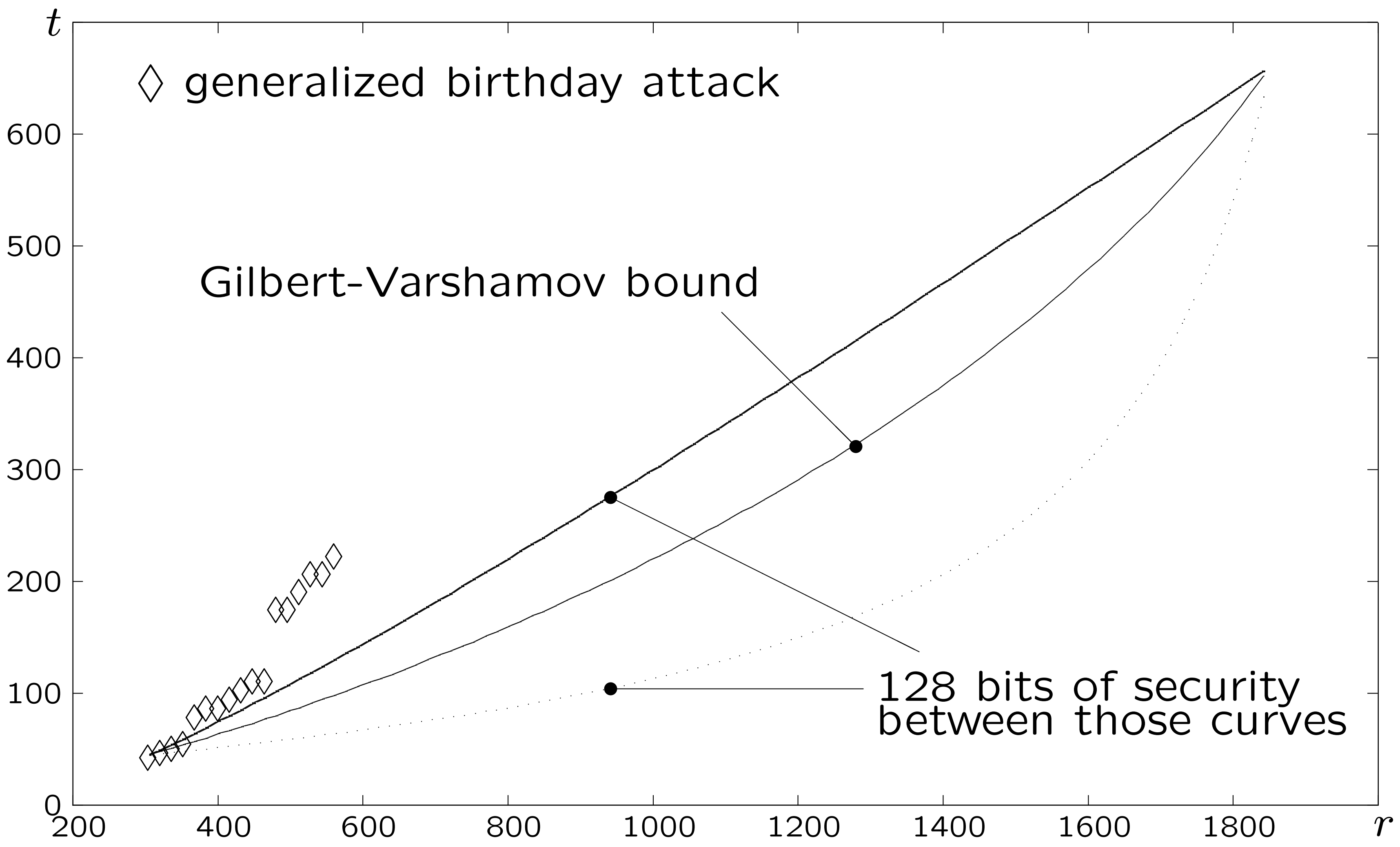
Decoding attack for $n = 1024$ and security 2^{128}



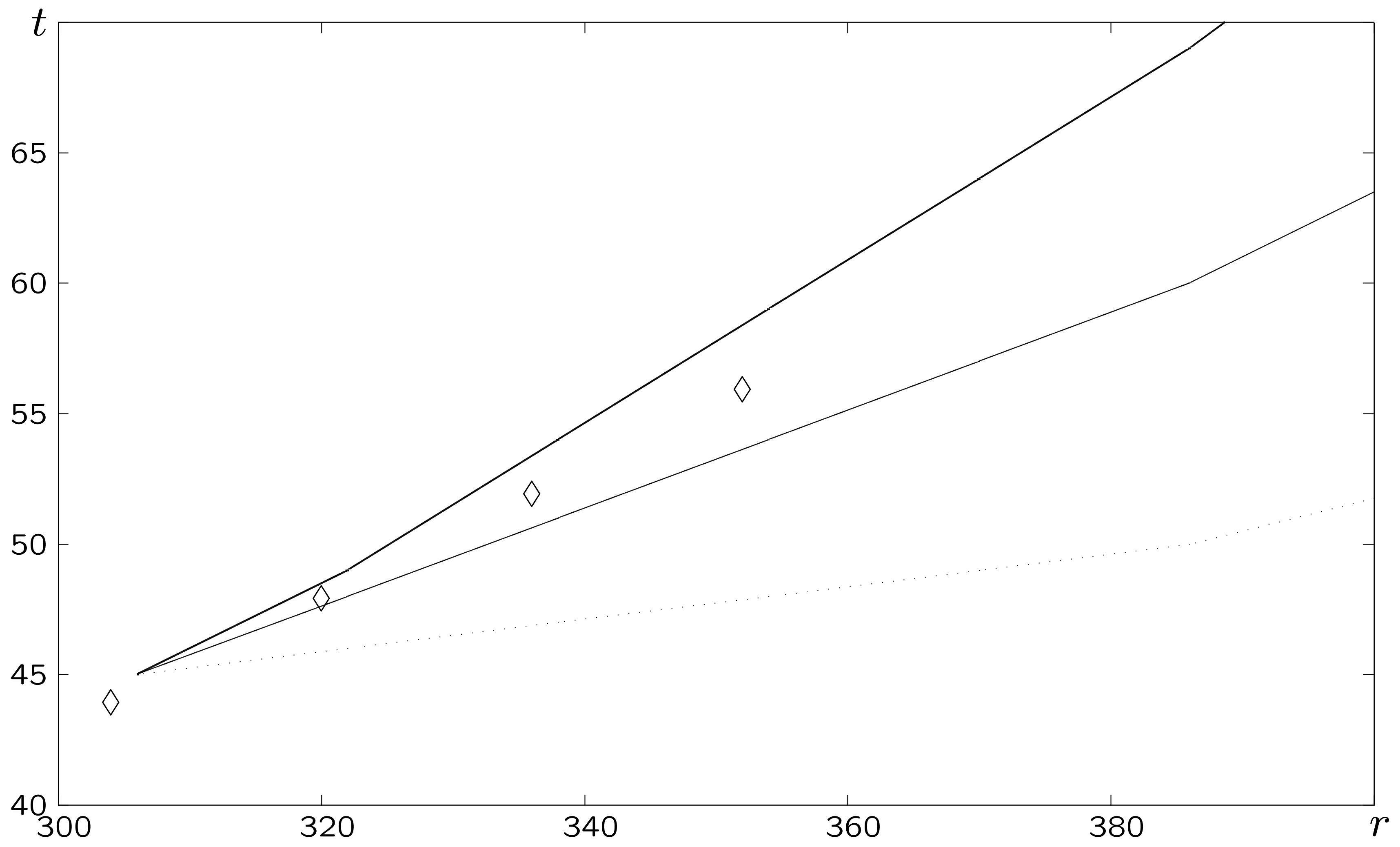
Decoding attack for $n = 1024$ and security 2^{128} – Zoom



Decoding attack for $n = 2048$ and security 2^{128}



Decoding attack for $n = 2048$ and security 2^{128} – Detail



II.2 Encoding errors

Encoding errors

In practice there is an encoding problem and we need a mapping (preferably injective) $\theta : \{0,1\}^\ell \to W_{n,t}$

- Fixed length and injective. Let $0 \leq i_1 < i_2 < \ldots < i_t < n$

  $$\begin{array}{rccc} S: & W_{n,t} & \longrightarrow & \left[0, \binom{n}{t}\right[\\ & (i_1, \ldots, i_t) & \longmapsto & \binom{i_1}{1} + \ldots + \binom{i_t}{t} \end{array}$$

  From this we can construct an injective mapping $\{0,1\}^\ell \to W_{n,t}$ with $\ell = \left\lfloor \log_2 \binom{n}{t} \right\rfloor$ and complexity quadratic in ℓ.

- Variable length and bijective. We can define an (unambiguous) encoding $\{0,1\}^* \to W_{n,t}$ with linear complexity and an input average length very close to ℓ

- Other trade-offs (regular words, . . .)

Regular words

The word is divided as evenly as possible into n/t part, each of them will have weight one exactly. We denote $R_{n,t}$ this set. Of course $R_{n,t} \subset W_{n,t}$.

$\cdots$ 1 $\cdots$	$\cdots$ 1 $\cdots$	$\cdots$ 1 $\cdots$	$\cdots$ 1 $\cdots$	$\cdots$ 1 $\cdots$	$\cdots$ 1 $\cdots$	$\cdots$ 1 $\cdots$

If n/t is an integer there are precisely $(n/t)^t$ regular words.

If $n/t = 2^b$ is a power of 2, then $|R_{n,t}| = 2^{bt}$ and the encoding $\{0,1\}^{bt}$ is particularly easy

$$\begin{array}{ccccc} \{0,1\}^{bt} & \longrightarrow & [0, 2^b[^t & \longrightarrow & R_{n,t} \\ & & (j_1, j_2 \ldots, j_t) & \longmapsto & (j_1, j_2 + 2^b, \ldots, j_t + 2^{b(t-1)}) \end{array}$$

The security of regular words

Syndrome decoding for regular words is NP-complete (Finiasz, 2004).

We have $|W_{n,t}| = \binom{n}{t} \approx n^t/t!$ and $|R_{n,t}| = n^t/t^t$. The ratio is $\approx$ exp(t), so decoding a regular error of weight t can be easier by a factor at most exp(t).

In practice, decoding attack have the same cost when $t \leq t_0$ and are more expensive for regular words when t gets larger.

For larger values of t the generalized birthday attack is not much more expensive for regular word, so it often becomes the best attack.

What have we got so far ?

We have got a mean to produce efficient mappings $f : \{0,1\}^\ell \to \{0,1\}^r$ whose security is reduced to instances of syndrome decoding.

We have a mean to evaluate the "practical" security of those mappings.

We will now consider more precisely two cases

- $\ell = r$ with which we can design stream ciphers.
- $\ell > r$ with which we can design hash functions.

III. New designs

How does this relates to the McEliece encryption scheme ?

McEliece encryption scheme uses a binary code $\mathcal{C}$ of length n and dimension k. The public key is a generator $(k \times n)$ matrix G of $\mathcal{C}$ and the encryption mapping is the following

$$\begin{array}{ccccc} \{0,1\}^k & \longrightarrow & \mathcal{C} & \longrightarrow & \{0,1\}^n \\ m & \longmapsto & x = mG & \longmapsto & x + e \end{array}$$

where the error e is chosen randomly of weight t.

The trapdoor is a t-error correcting procedure for $\mathcal{C}$.

Typical sizes for 80 bits of security are

$$n = 2048, k = 1696, r = 352, t = 32$$

How does this relates to the Niederreiter encryption scheme ?

Niederreiter encryption scheme uses a binary code $\mathcal{C}$ of length n and codimension r. The public key is a parity check $(r \times n)$ matrix H of $\mathcal{C}$ and the encryption mapping is the following

$$\begin{array}{cccccc} (\{0,1\}^{\ell} & \longrightarrow) & W_{n,t} & \longrightarrow & \{0,1\}^{n} \\ (m & \longmapsto) & e & \longmapsto & eH^{T} \end{array}$$

The trapdoor is a t-error correcting procedure for $\mathcal{C}$.

Typical sizes for 80 bits of security are

$$n = 2048, k = 1696, r = 352, t = 32$$

III.1 Reducing the matrix size

Reducing the matrix size

One of the drawbacks of code-based mappings is that they require a large binary matrix (can be several Mbits).

In public key cryptography it is difficult to overcome that problem (there is an attempt by Gaborit, though).

For one-way functions (without trapdoor), the matrix is random, so with have options:

- use a pseudo-random number generator, so we only need to know a seed,
- use a structured matrix (cyclic or quasi-cyclic for instance), so we only need to know the first row.

Block circulant matrices

A circulant square matrix is composed of all the cyclic shifts of a single word.

A block circulant matrix is obtained by concatenating several circulant square matrices (R_i)

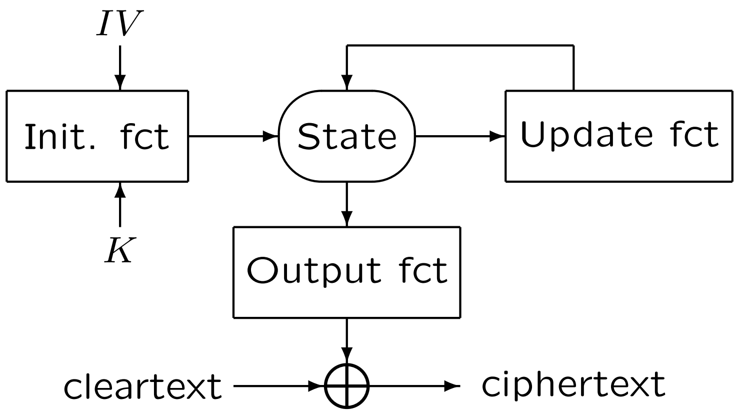
$$H = \begin{array}{|c|c|c|c|}\hline R_1 & R_2 & \cdots & R_s \\ \hline\end{array}$$

The code defined by H is quasi-cyclic. The syndrome mapping is not likely to be easier to solve for quasi-cyclic codes.

The Holy Grail of coding theory is a class of good block codes (quasi-cyclic codes meet the GV bound, which mean "good" in coding theory) which has an efficient complete decoder (i.e. the syndrome mapping can be inverted everywhere).

The SYND stream cipher

Joint work with Ph. Gaborit and C. Lauradoux (ISIT, june 2007)



Basic idea (QUAD): one can securely extract more than $\log n$ bits after each update of n bits state.

In fact, if the fonction allows it n (or more) bits can be extracted each round.

The state update function will be a syndrome mapping with same size of input and output (we choose the same size for the output).

We use regular word encoding for efficiency.

The SYND stream cipher – Performances

t	security	n	r	key size	cycle/byte
16	64	4096	128	64	22
24	96	6144	192	96	46
32	**128**	**8192**	**256**	**128**	**27**
48	192	12288	384	192	47
64	256	16384	512	256	53
128	512	32768	1024	512	83
AES-CTR	128	-	-	128	26

The security is given by a search in the key space, the other attacks (decoding, birthday) are not faster.

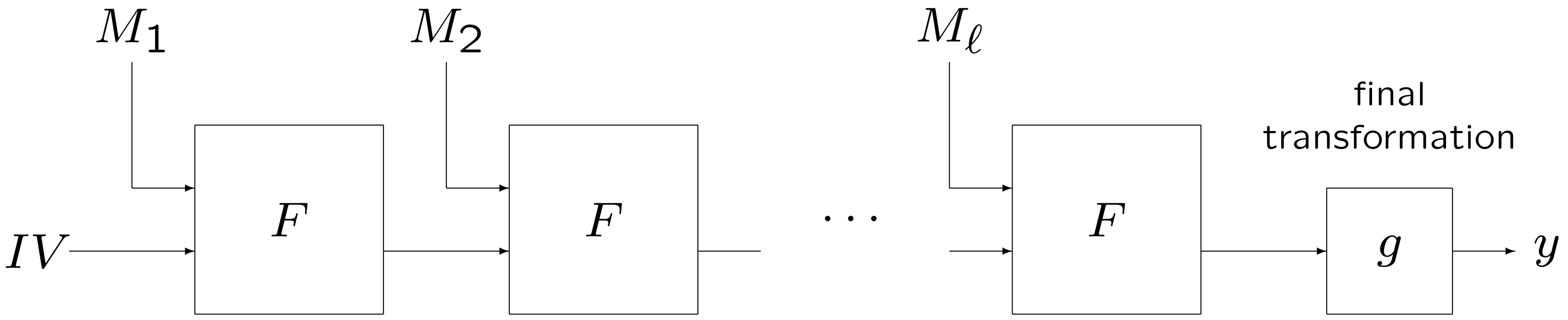
The speed is comparable with the AES in counter mode.

We have security reduction (loose).

With circulant matrices, we can lower the memory requirements.

Hash function

Joint work with D. Augot, M. Finiasz, Ph. Gaborit



The compression function F is a syndrome mapping. The final transformation g is any truncated one-way function, but NOT a syndrome mapping.

Hash function (2)

In the case of a syndrome-based compression fonction, the generalized birthday attack is very efficient. As a consequence, the state (size of the output of F) is large (512 or preferably 1024 bits).

- The final transformation becomes necessary, or we cannot achieve $n/2$ security with n bits of output.
- The key reduction techniques are particularly important (several dozens of Mbits for the matrix).

Hash function – Security

For a hash function with n bits of output, the security requirements are

- n bits of security against first and second preimage attacks,
- $n/2$ bits of security against collision attacks.

All those attacks can be reduced to the inversion of some syndrome function. Finding a collision for a syndrome mapping $W_{n,t} \rightarrow \{0,1\}^r$ is not harder than inverting another syndrome mapping $W_{n,2t} \rightarrow \{0,1\}^r$.

Hash function – Parameters and performances

security	r	t	n	cycles/byte
64	512	512	131 072	90
	512	450	230 400	165
	1 024	2^{17}	2^{25}	340
80	512	170	43 520	281
	512	144	73 728	240
128	1 024	1 024	262 144	121
	1 024	904	462 848	371
	1 024	816	835 584	162

The cost of the final transformation is ommitted (but is negligible for large messages).

The fastest version with 128 bits of security is two times slower than SHA256.

Conclusions

- The syndrome mapping is a secure and efficient one-way primitive.
- Its flexibility allows many applications in secret key cryptography.
- Generalized birthday attack and its application to decoding need to be studied further.
- There are other (bad?) properties that were not mentionned here. For instance malleability: one can easily find three distinct inputs with related outputs.