Cryptanalysis of the Sidelnikov cryptosystem

Lorenz Minder, Amin Shokrollahi

{lorenz.minder,amin.shokrollahi}@epfl.ch.

LMA, EPFL

McEliece type cryptosystems

PKCS based on error-correcting codes. $\mathcal{C}$: error-correcting code.

$$\text{Encryption} \leftrightarrow \text{Encode with } \mathcal{C} \text{ and add errors}$$

$$\text{Decryption} \leftrightarrow \text{Decode noisy codewords from } \mathcal{C}$$

Linear codes

- have a short description (basis of a linear space),
- are easy to encode (linear map),
- are hard to decode in general, but efficiently decodable codes exist.

Can decodeable codes be disguised?

Disguising linear codes

$\mathcal{C}$ is an $[n, k]$ binary linear code with $k \times n$ generator matrix G, correcting t errors.

- Pick a random basis of the vector space.
  ($G \mapsto A \cdot G$, where A is $k \times k$ random invertible.)
- Permute coordinate positions.
  Notation: $\mathcal{C}^\sigma$ is $\mathcal{C}$ with σ applied to its coordinate positions.
  ($G \mapsto G \cdot P$, where P is an $n \times n$ permutation matrix for σ.)

So, $G_{\text{pub}} := AGP$ is a disguised generator matrix for $\mathcal{C}^\sigma$.

McEliece type cryptosystems

- Public key: G_{pub} and t.
- Encryption: The binary vector $x = (x_1, \dots, x_k)$ is encrypted as

$$y := xG_{\text{pub}} + e \in \mathbb{F}_2^n,$$

  where e is a random, weight t error pattern.
- Private key: Decoder for $\mathcal{C}^\sigma$.
- Decryption: Decode.
- Hardness assumptions:
  - Decoding is hard in general.
  - Recovering the structure of $\mathcal{C}^\sigma$ is hard.

How secure is it ?

It depends on the code. Different families have been considered:

- *Goppa*-codes, originally proposed by McEliece, 1978. Unbroken.
- *Reed-Solomon*-codes proposed by Niederreiter, 1986. Broken by Sidelnikov & Shestakov, 1992
- *Reed-Muller*-codes proposed by Sidelnikov, 1994. Our target.
- *Algebraic-Geometry*-codes proposed by Janwa & Moreno, 1995.
- Non-algebraic codes. Usually easy to break.

Why Reed-Muller Codes ?

Reed-Muller codes were proposed, because:

- Resulting public keys are small.
- Can decode many more than $d/2$ errors with high probability (d is the minimum distance).
  - Thwarts direct decoding attacks.
  - Improves information rate.
- The decoder is very fast.

Our goal

We are given r, m and a random basis of a permuted rth order Reed-Muller code of length 2^m, $\mathcal{R}(r,m)^\sigma$, that is, a matrix $G_{\text{pub}} = AGP$. We want to find a permutation τ such that

$$\mathcal{R}(r,m)^{\tau \circ \sigma} = \mathcal{R}(r,m).$$

Want a private key for a given public key.
In general, $\tau \circ \sigma \neq id$.

Reed-Muller Codes

f	codeword							
1	1	1	1	1	1	1	1	1
v_1	0	0	0	0	1	1	1	1
v_2	0	0	1	1	0	0	1	1
v_3	0	1	0	1	0	1	0	1
$v_2 v_1$	0	0	0	0	0	0	1	1
$v_1 v_3$	0	0	0	0	0	1	0	1
$v_3 v_2$	0	0	0	1	0	0	0	1

- $(\mathbb{F}_2[v_1, \ldots, v_m]/v_1^2 - v_1, \ldots, v_m^2 - v_m)_{\leq r}$
- $\mathcal{R}(r, m)$: all evaluations on all points, $v_i \in \mathbb{F}_2$.
- $n = 2^m$, $k = \sum_{i=0}^{r} \binom{m}{i}$, $d = 2^{m-r}$.

Minimum weight words

Boolean functions which are r linearly independent affine factors generate minimum weight words. E.g.,

$$f = v_1 v_2 \cdots v_r.$$

Is there any other way to construct minimum weight words? No. We have (Kasami & Tokura):

Proposition. If $f(v_1, \ldots, v_m)$ generates a minimum weight word in $\mathcal{R}(r, m)$, then f can be written as

$$f = f_1 \cdots f_r,$$

where the f_i are affine functions of $v_1, \ldots, v_m$.

Exploiting minimum weight words

Sketch of the procedure:

- Find a minimum weight word. (E.g., use the Canteaut-Chabaud algorithm.)
- Split a factor of the word. The factor will lie in $\mathcal{R}(r-1,m)^{\sigma}$.
- Repeat until a basis of $\mathcal{R}(r-1,m)^{\sigma}$ has been found.
- Repeat until a basis of $\mathcal{R}(1,m)^{\sigma}$ has been found.
- Identify τ such that

$$\mathcal{R}(1,m)^{\tau\circ\sigma} = \mathcal{R}(1,m).$$

Then $\mathcal{R}(r,m)^{\tau\circ\sigma} = \mathcal{R}(r,m)$.

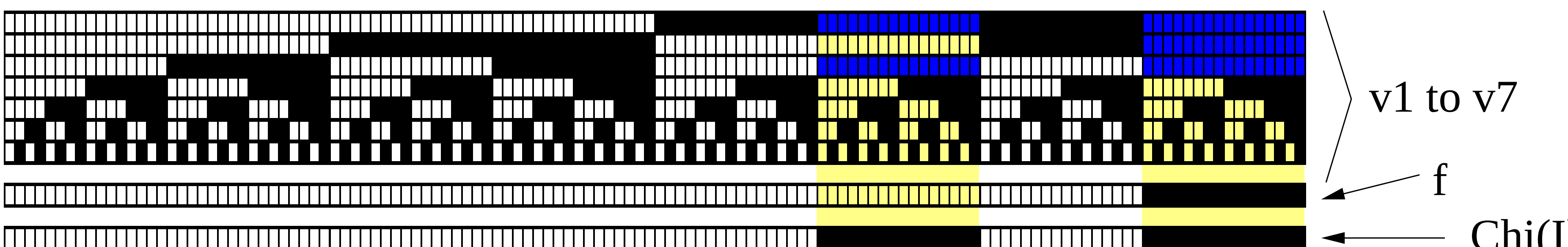
Factoring minimum weight words

f: minimum weight word. W. l. o. g., $f = v_1 \cdots v_r$.

Let $(k_1, \ldots, k_r) \in \mathbb{F}_2^r \setminus \{\hat{1}\}$. Consider

$$I := \underbrace{\{v_1 = 1, \ldots, v_r = 1\}}_{\mathrm{supp}(f)} \cup \{v_1 = k_1, \ldots, v_r = k_r\}.$$

Example. $\mathcal{R}(3,7), f = v_1 v_2 v_3, k = (1,0,1)$.



In this case $\chi_I = v_1 v_3 \in \mathcal{R}(2, 7)$.

Factoring minweight words (cont'd)

From the last slide:

$$I := \{v_1 = 1, \ldots, v_r = 1\} \cup \{v_1 = k_1, \ldots, v_r = k_r\}.$$

W.l.o.g., if $k = (\underbrace{1, \ldots, 1}_{t \text{ times}}, 0, \ldots, 0)$, then

$$\chi_I = v_1 \cdots v_t \cdot (1 + v_{t+1} + v_{t+2}) \cdots (1 + v_{r-1} + v_r).$$

Therefore $\deg(\chi_I) \leq r - 1$ and so $\chi_I \in \mathcal{R}(r-1, m)$.

$\Longrightarrow$ want to explicitly construct a χ_I.

$\Longrightarrow$ have to compute a set I given f.

Finding a set I

$\mathcal{C}_{\mathrm{supp}(f)}$ is $\mathcal{R}(r,m)^\sigma$ shortened on $\mathrm{supp}(f)$.

It can be shown that, up to symbol permutation,

$$\mathcal{C}_{\mathrm{supp}(f)} \subseteq \mathcal{R}(r-1,m-r) \times \cdots \times \mathcal{R}(r-1,m-r),$$

with each of the factors in the cartesian product lying on the sets $\{v_1 = k_1, \ldots, v_r = k_r\}$, each factor for a different k.

Identifying the sets $\{v_1 = k_1, \ldots, v_r = k_r\}$ is the same as identifying the positions of the (“inner”) $\mathcal{R}(r-1,m-r)$-blocks.

Finding inner words

Use Sendrier's algorithm for concatenated codes:

- Show that the support of any minimum weight word in $\mathcal{C}^{\perp}_{\mathrm{supp}(f)}$ is contained within a single inner word.
- Let $x \in \mathcal{C}^{\perp}_{\mathrm{supp}(f)}$ be of minimum weight. If $x_i = 1 = x_j$, then i and j are positions in the same inner block.
- Collect enough such witnesses.

Recap

The steps to find a vector in $\mathcal{R}(r-1,m)^\sigma$ are:

- Find a minimum weight word f in $\mathcal{C} = \mathcal{R}(r,m)^\sigma$.
- Compute the shortened code $\mathcal{C}_{\mathrm{supp}(f)} \subset \mathcal{C}$.
- Recover the cartesian product structure of $\mathcal{C}_{\mathrm{supp}(f)}$.

If S is the set of positions of any inner word in $\mathcal{C}_{\mathrm{supp}(f)}$, the word with ones on the set

$$S \cup \mathrm{supp}(f)$$

is a word in $\mathcal{R}(r-1,m)^\sigma$.

Finishing up

By iteration, we construct

$$\mathcal{R}(r,m)^\sigma \supset \mathcal{R}(r-1,m)^\sigma \supset \cdots \supset \mathcal{R}(1,m)^\sigma.$$

Since $\mathcal{R}(r,m)^\sigma$ can be uniquely constructed from $\mathcal{R}(1,m)^\sigma$, need to solve the problem for $\mathcal{R}(1,m)^\sigma$, i.e., need to

find a permutation τ, such that

$$\mathcal{R}(1,m)^{\tau\circ\sigma} = \mathcal{R}(1,m).$$

Recovering $\mathcal{R}(1,m)^\sigma$

f	codeword															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
v_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
v_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
v_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
v_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
col	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

- Column index $\leftrightarrow$ binary value $(v_m v_{m-1} \cdots v_1)_2$.
- G: random generator of $\mathcal{R}(1,m)^\sigma$. Throw away one row, and identify a permutation by the values of the columns. Success probability: 1/2.

How practical is it?

Running times on PC:

	$r = 2$	$r = 3$	$r = 4$
$m = 7$ $(n = 128)$	0.009s	0.03s	
$m = 8$ $(n = 256)$	0.04s	0.18s	
$m = 9$ $(n = 512)$	0.24s	1.26s	2m 57s
$m = 10$ $(n = 1024)$	1.77s	16.15s	22h 49m 57s
$m = 11$ $(n = 2048)$	12.14s	5m 20.8s	10d 11h 55m

- It is practical whenever it is practical to find minimum weight words.
- Performance degrades if r is large.
- For large r, Reed-Muller codes are not useful.