Low-weight correlation-immune Boolean functions for counter-measures to side channel attacks

Claude Carlet

LAGA, Universities of Paris 8 and Paris 13, CNRS, France

and University of Bergen, Norway

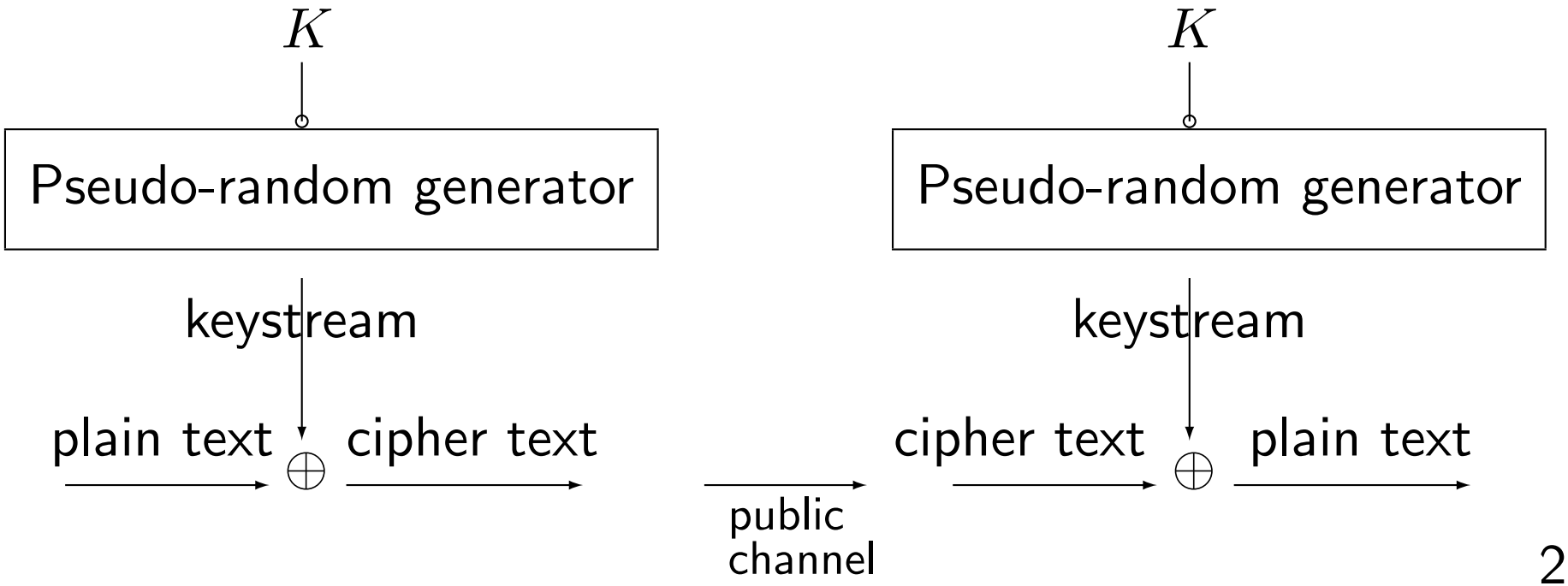
Work in common with Xi Chen

Outline

- Correlation immune functions in the framework of stream ciphers
- Side Channel Attacks and their counter-measures
- How Boolean functions play a new role in this framework
- Why this poses new questions on correlation-immune Boolean functions
- What is known on minimum weight CI functions
- Constructions of low weight CI Boolean functions

Correlation immune functions in the framework of stream ciphers

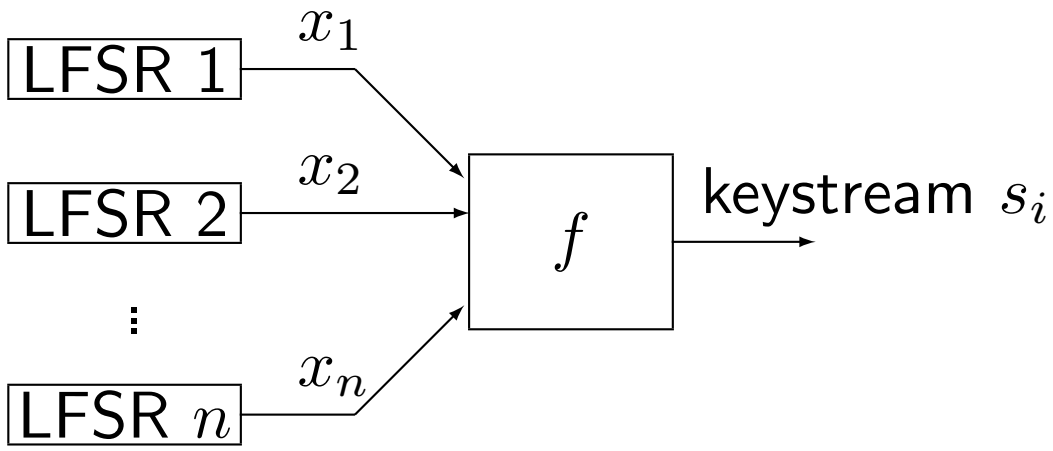
Synchronous stream ciphers :



Every pseudo-random generator (PRG) consists in a linear part (for efficiency) and a nonlinear part (for robustness).

Boolean functions $f : \mathbb{F}_2^n \to \mathbb{F}_2$ are often used in the nonlinear part.

A classical *model* for their use combines the outputs of several Linear Feedback Shift Registers (LFSR) is the *combiner model* :



Several attacks exist on this model, among which a divide and conquer attack called the *Siegenthaler correlation attack*.

To withstand it, f must have no correlation with any subset of at most m variables, where m is as high as possible.

• Equivalent definition : the output distribution of f should not change when at most m input variables are fixed.
We say then that f is *correlation-immune* of order m (m-CI).

• Characterization by the *Walsh transform* (Xiao-Massey) :

$$\forall a \in \mathbb{F}_2^n,\ 1 \leq w_H(a) \leq m \Rightarrow W_f(a) = \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+a\cdot x} = 0,$$

where w_H is the Hamming weight and "$\cdot$" the usual inner product in $\mathbb{F}_2^n$.

• Characterization by the *Fourier-Hadamard transform* :

$$\forall a \in \mathbb{F}_2^n,\ 1 \leq w_H(a) \leq m \Rightarrow \widehat{f}(a) = \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{a \cdot x} = 0,$$

since $W_f(a) = -2\widehat{f}(a)$.

• Characterization by (nonlinear) *codes* : the code C equal to the support $\{(x \in \mathbb{F}_2^n \,|\, f(x) = 1\}$ of f has dual distance at least $m + 1$.

Recall : given a code $C \subseteq \mathbb{F}_2^n$, the distance enumerator of C is

$$D_C(X, Y) = \frac{1}{|C|} \sum_{(u,v) \in C^2} X^{n - d_H(u,v)} Y^{d_H(u,v)}.$$

The dual distance of C is the minimal nonzero degree of Y in the monomials with nonzero coefficients in $D_C(X+Y, X-Y)$.

• Characterization by *orthogonal arrays* : the $|C| \times n$ array of all elements of C is an orthogonal array (with no repetition) of strength m.

In practice, functions for the combiner model need to be m-CI and balanced (that is, m-resilient) for sufficiently large m and also highly nonlinear with algebraic degree as high as possible.

The nonlinearity $nl(f)$ of a function f is the minimum Hamming distance between f and affine functions.

Its algebraic degree $d_{alg}(f)$ is the degree of its Algebraic Normal Form (ANF)

$$f(x_1, \cdots, x_n) = \bigoplus_{I \subseteq \{1,\ldots,n\}} a_I \left(\prod_{i \in I} x_i \right).$$

In 2003 came *algebraic attacks* and more problematic *fast algebraic attacks* (FAA).

To resist FAA, there should not exist $g \neq 0$ such that $d_{alg}(g)$ is small and $d_{alg}(fg)$ is not large.

Then, if $d_{alg}(f)$ is not large, f does not resist FAA (since the attacker can take $g = 1$).

Weakness of CI functions for stream ciphers :

Correlation immune functions have low algebraic degrees :

$$d_{alg}(f) \leq n - m.$$

Correlation immune functions are then weak against :

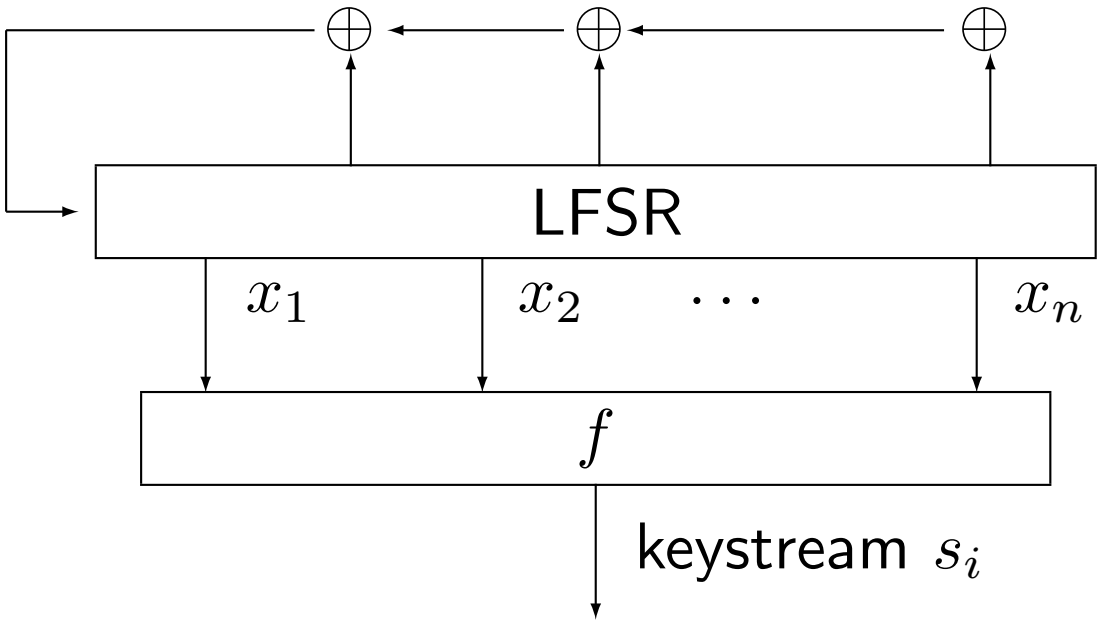
- the Berlekamp-Massey attack, whose complexity is nowadays slightly more than linear in $L^{d_{alg}(f)}$, where L is the average size of the LFSRs,

- the Ronjom-Helleseth attack, whose complexity is linear in $\binom{nL}{d_{alg}(f)}$,

- the fast algebraic attack, whose complexity can be also very low when f has not high algebraic degree.

Consequence : another model is preferred which does not need high order correlation immunity : the filter model.

Filter model



End of the story for correlation-immune functions ?

Side Channel Attacks and their counter-measures

The implementation of cryptographic algorithms in devices like smart cards (mainly software), FPGA or ASIC (hardware) leaks information on the data manipulated by the algorithm, leading to *side channel attacks* (SCA).

The attacker model is then not a black box but a greay box.

This information can be *traces* of electromagnetic emanations, power consumption, photonic emission...


XILINX
5V
VCCINT
VCCO
VCCAUX
GND
VSE
SPI INTERFACE
VIRTEX-5 FF324 PROTO

SCA are very powerful on block ciphers if countermeasures are not included in the implementation of the cryptosystems, since they can use information on the data manipulated during the first round (which has not reached good diffusion).

A *sensitive variable* is chosen in the algorithm, whose value is stored in a *register* and depends on the plaintext and a few key bits.

The register *leaks*.

The emanations from the register are measured. They disclose a noisy version of a real-valued function $\mathcal{L}$ of the sensitive variable.

For instance, in the so-called *Hamming weight leakage model*, $\mathcal{L}(Z)$ equals the Hamming weight of Z.

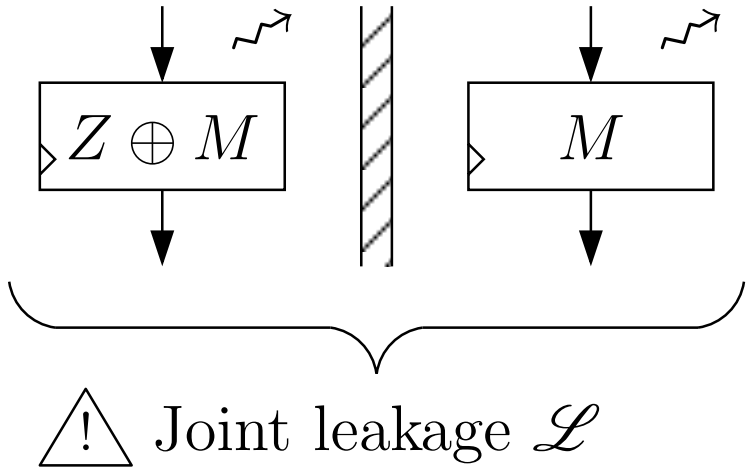
A statistical method finds then the value of the key bits which optimizes the correlation between the traces and a *modeled leakage*.

The original implementation of the AES can be attacked this way in a few seconds with a few traces.

Counter-measures fortunately exist.

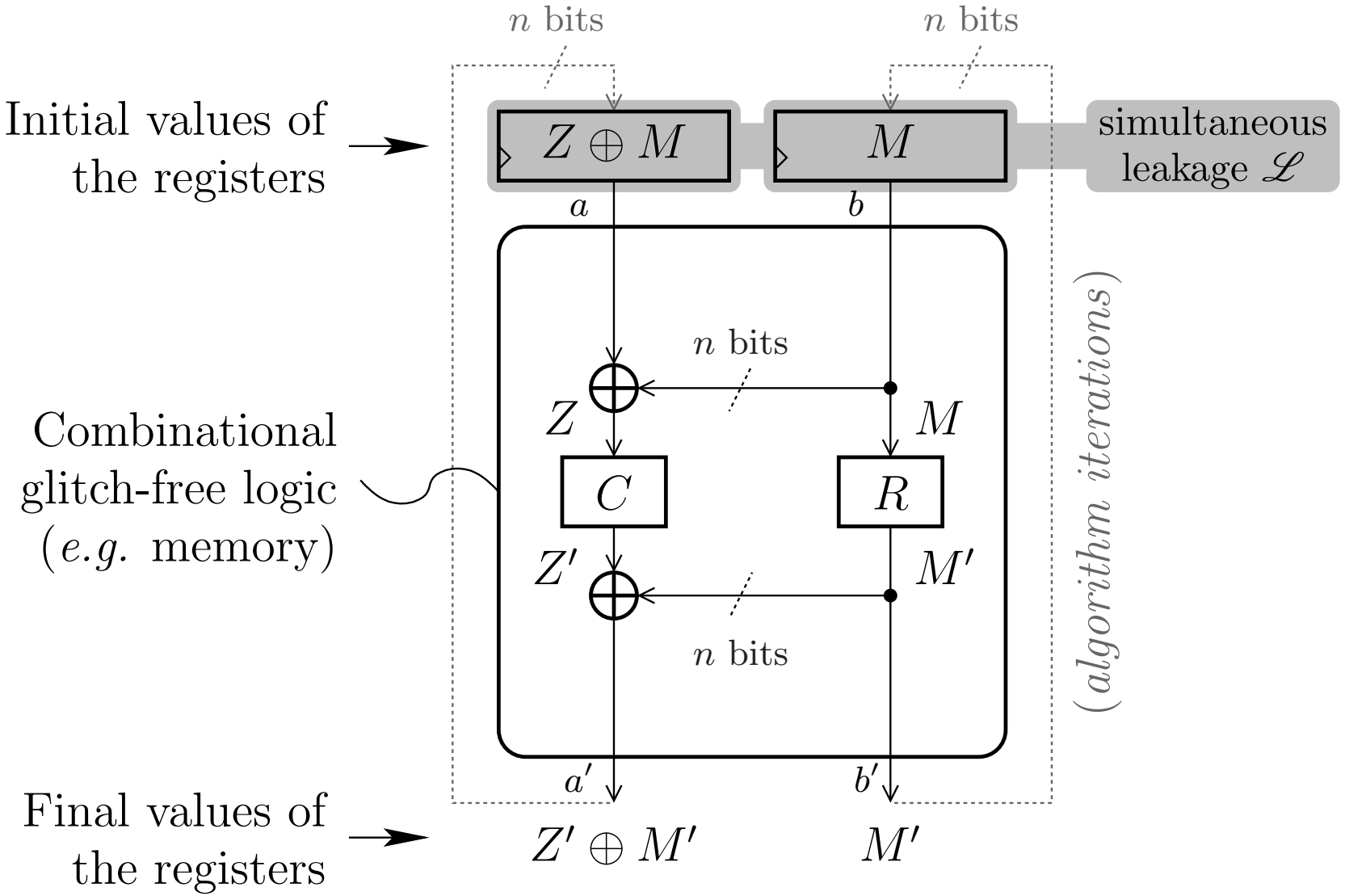
Most common : *mask* each sensitive variable Z by splitting it.

- 2 shares : $Z \oplus M \quad \| \quad M$, where M is drawn at random.



For going through boxes

In hardware (FPGA, ASIC, ...) :


Initial values of the registers
n bits
n bits
Z ⊕ M
M
simultaneous leakage ℒ
a
b
n bits
Z
M
Combinational glitch-free logic (e.g. memory)
C
R
Z′
M′
n bits
(algorithm iterations)
a′
b′
Final values of the registers
Z′ ⊕ M′
M′

In software (smart cards) : transform every function $x \mapsto F(x)$ in the algorithm into a function $F' : (m_0, m_1) \mapsto (m'_0, m'_1)$ such that :

$$m'_0 + m'_1 = F(m_0 + m_1)$$

(i.e. F' is a function on shares of x providing shares of $F(x)$) and the knowledge of one intermediate variable does not give any information on x.

Such F' is called a *masked version* of F.

Masking linear functions is costless but masking S-boxes has a cost.

In software applications (smart cards), masking the algorithm can multiply by more than 20 the execution time.

An AES runs in 3629 cycles without masking and in 100 000 with masking.

The program executable file size is also increased because all the rest of the computations on Z needs to be modified into computations on shares.

In hardware applications (ASIC, FPGA), the implementation area is roughly tripled.

Higher order attacks : The counter-measure of masking with a single mask (i.e. two shares) cannot resist *Higher order SCA* (HO-SCA) :

- The attacker starts with a first order attack, exploiting the leakage $\mathcal{L}(Z)$. This is successful if $\mathbb{E}(\mathcal{L}|Z = z)$ depends on z.

- if $\mathbb{E}(\mathcal{L}|Z = z)$ does not depend on z, then the attacker can try a second order attack, on $\mathcal{L}^2$ (or on the product of two leakages, which is more difficult in hardware but possible in software),

- if $\mathbb{E}(\mathcal{L}^2|Z = z)$ does not depend on z, then the attacker can increase the order of the attack until it is successful.

Higher order masking : d-th order masking allows resisting d-th order SCA :

$d+1$ shares : $M_1, \ldots, M_d$ are chosen at random and

$$M_{d+1} = Z \oplus M_1, \cdots \oplus M_d.$$

The complexity of the HO-SCA attack (in time and in the number of traces) is exponential in the order : $O(V^d)$, where V is the variance of the noise (indeed, raising the leakage at the d-th power raises the noise at the d-th power).

The cost in terms of running time and of memory is quadratic in d.

Hence, theoretically, the designer can take advantage over the attacker.

However, an advantage of the attacker over the designer is that the implementation must be efficient today while the SCA can be performed in the future.

Hence it is very important to be able to reduce the cost of counter-measures against SCA.

How Boolean functions play a new role in this framework

- Leakage squeezing (hardware)

At first order, the pair (M_0, M_1) such that $M_0 + M_1 = Z$ is not processed as is in the device, but in the form of $(M_0, F(M_1))$.

Efficiency of leakage-squeezing for first-order :

Theorem The first-order leakage squeezing counter-measure with a permutation F resists the attack of order d if and only if :

$$\forall a, b \in \mathbb{F}_2^n,\ 1 \leq w_H(a) + w_H(b) \leq d \Rightarrow \sum_{x \in \mathbb{F}_2^n} (-1)^{b \cdot F(x) + a \cdot x} = 0,$$

that is, the indicator (characteristic function) of the graph $\mathcal{G}_F = \{(x, F(x), x \in \mathbb{F}_2^n\}$ of F is d-CI.

Equivalently, the code $\mathcal{G}_F = \{(x, F(x), x \in \mathbb{F}_2^n\}$ has dual distance at least $d + 1$.

This code is in general nonlinear ; it is linear when F is linear.

Such a code $\mathcal{G}_F = \{(x, F(x), x \in \mathbb{F}_2^n\}$, where F is a permutation, admits $\{1, \ldots, n\}$ and $\{n + 1, \ldots, 2n\}$ as information sets.

Recall : an information set for a code is a set I of indices such that every possible tuple of length $|I|$ occurs in exactly one codeword within the specified coordinates $x_i;\ i \in I$.
Every linear code is systematic.

A $[n = 2k, k]$ code having two information sets complementary of each other is called a *Complementary Information Set* (CIS) code.

The CIS codes with best dual distances have been investigated in 2012 for $n \leq 65$ by C.C., P. Gaborit, J.-L. Kim, and P. Solé.

Some CIS codes with best dual distance are linear, some are not :

for $n = 4$ the best dual distance is 4, achieved by a linear code

for $n = 8$ (AES) the best dual distance is 6, achieved by a nonlinear code : the Nordstrom-Robinson code, that is, the Kerdock code of length 16 (the best linear code gives 5).

Efficiency of leakage squeezing for second order :

$Z = M_0 + M_1 + M_2$ and $(M_0, F_1(M_1), F_2(M_2))$ is processed.

Theorem The second-order leakage squeezing counter-measure with permutations F_1, F_2 resists the SCA of order d if and only if :

$$\forall (a, b, c),\ a \neq 0,\ (w_H(a) + w_H(b) + w_H(c) \leq d) \Rightarrow$$

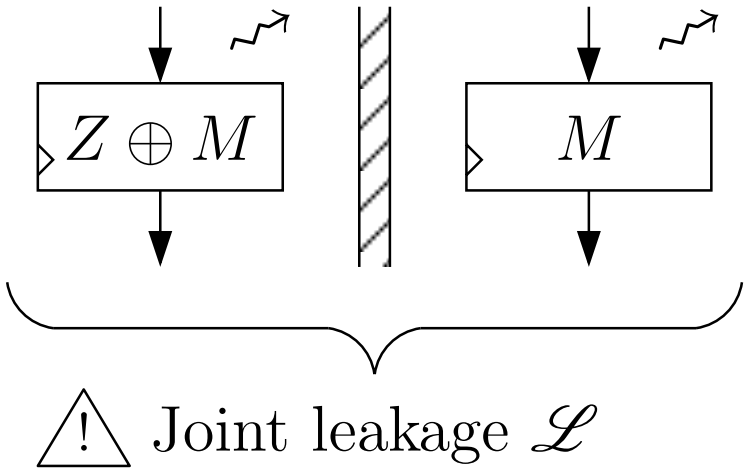
$$\sum_{x \in \mathbb{F}_2^n} (-1)^{b \cdot F_1(x) + a \cdot x} = 0 \text{ or } \sum_{x \in \mathbb{F}_2^n} (-1)^{b \cdot F_2(x) + a \cdot x} = 0.$$

Equivalently, the code $\mathcal{G}_{F_1,F_2} = \{(x+y, F_1(x), F_2(y)) |\, x,\, y \in \mathbb{F}_2^n\}$ has dual distance at least $d+1$.

Such codes have been studied by C.C., F. Freibert, S. Guilley, M. Kiermaier, J.-L. Kim and P. Solé.

- Rotating S-boxes Masking (RSM, hardware)

  To avoid the joint leakage :



which allows high-order SCA, the mask M is not processed at all.

Instead, the computation for the next S-box is done with a Look-Up-Table (LUT) of the masked S-box $S'(x) = S(x \oplus M) \oplus M'$.

This allows a perfect protection against SCA.

But having a LUT for each masked version of each S-box is not possible for reasons of memory.

A small number of S-boxes (*e.g.* $w = 16$ for the AES) are then embedded already masked in the implementation and evaluated in parallel (especially relevant for the ciphers that use many instances of the same S-box, *e.g.* AES or PRESENT).

At every encryption, the allocation of the S-box for each of the 16 plaintext bytes is done randomly.

This counter-measure can then be attacked by a high order SCA.

Theorem The countermeasure resists the d-th order attack if and only if the indicator f of the mask set satisfies

$$\forall a \in \mathbb{F}_2^n,\ 1 \leq w_H(a) \leq d \Rightarrow \sum_{x \in \mathbb{F}_2^n} (-1)^{f(x)+a\cdot x} = 0,$$

that is, the indicator of $\mathcal{M}$ is a d-CI function.

Equivalently, the mask set is a code of dual distance at least $d+1$.

For d as large as possible, we look for such functions of *minimum nonzero Hamming weight*, since the lower the weight of this function, the cheaper the countermeasure.

Why this poses new questions on correlation-immune Boolean functions

Known constructions allow constructing balanced CI (resilient) functions but not low weight CI-functions. For instance :

1. With Maiorana McFarland construction :

$$f(x,y) = x \cdot \phi(y) \oplus g(y)\,;\ x \in \mathbb{F}_2^r,\ y \in \mathbb{F}_2^{n-r},$$

we have

$$W_f(a,b) = \sum_{y \in \phi^{-1}(a)} (-1)^{g(y) \oplus b \cdot x},$$

and for $\phi^{-1}(0) \neq \emptyset$, it is hard handling $a = 0$ and $w_H(b) \leq m$.

2. With indirect sum :

$$h(x,y) = f_1(x) \oplus g_1(y) \oplus (f_1(x) \oplus f_2(x))(g_1(y) \oplus g_2(y)),$$

we have

$$W_h(a,b) =$$

$$\frac{1}{2} W_{f_1}(a) \left[W_{g_1}(b) + W_{g_2}(b) \right] + \frac{1}{2} W_{f_2}(a) \left[W_{g_1}(b) - W_{g_2}(b) \right],$$

and handling $a = 0$ and $w_H(b) \leq m$ is hard too, as well as $b = 0$ and $w_H(a) \leq m$.

What is known on minimum weight CI functions

$\omega_{n,d}$: minimum weight of CI functions of order d.

2^d divides $\omega_{n,d}$.

If $n \geq d \geq 1$, then

$$\omega_{n+1,d} \leq 2\omega_{n,d} \leq \omega_{n+1,d+1}.$$

Sketch of proof :
$g(x, x_{n+1}) = f(x)$;
$f(x) = g(x, 0)$.

TABLE 1: Lower bound on $\omega_{n,d}$ by the Delsarte LP bound

n \ d	1	2	3	4	5	6	7	8	9	10	11	12	13
1	2												
2	2	4											
3	2	4	8										
4	2	6	8	16									
5	2	8	12	16	32								
6	2	8	16	32	32	64							
7	2	8	16	48	64	64	128						
8	2	10	16	64	88	112	128	256					
9	2	12	20	96	128	192	224	256	512				
10	2	12	24	96	192	320	384	512	512	1024			
11	2	12	24	96	192	512	640	1024	1024	1024	2048		
12	2	14	24	112	176	768	1024	1536	1792	2048	2048	4096	
13	2	16	28	128	224	1024	1536	2560	3072	3584	4096	4096	8192

n \ d	1	2	3	4	5	6	7	8	9	10	11	12	13
1	**2**												
2	**2**	**4**											
3	**2**	**4**	**8**										
4	**2**	**8**	**8**	**16**									
5	**2**	**8**	**16**	**16**	**32**								
6	**2**	**8**	**16**	**32**	**32**	**64**							
7	**2**	**8**	**16**	**64**	**64**	**64**	**128**						
8	**2**	**12**	**16**	**64**	**128**	**128**	**128**	**256**					
9	**2**	**12**	**24**	**128**	**128**	**256**	**256**	**256**	**512**				
10	**2**	**12**	**24**	**128**	**256**	*512*	*512*	*512*	*512*	*1024*			
11	**2**	**12**	**24**	?	?	*512*	*1024*	*1024*	*1024*	*1024*	*2048*		
12	**2**	**16**	**24**	?	?	?	*1024*	*2048*	*2048*	*2048*	*2048*	*4096*	
13	**2**	**16**	*32*	?	?	?	?	*4096*	*4096*	*4096*	*4096*	*4096*	*8192*

Minimal value $\omega_{n,d}$ of the cardinal of $supp(f)$, where $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is d-CI.

The entries in bold have been obtained by using Satisfiability Modulo Theory (SMT) tools.

The entries in italic are obtained thanks to mathematical bounds.

For the entries with ? we have only upper and lower bounds.

Open question : the columns are they non-decreasing ?

Consequence : A byte-oriented block cipher (AES) can be protected with only 16 mask values against attacks of orders 1, 2 and 3.

Constructions of low weight CI Boolean functions

It is enough to deal with d even :

Proposition Let d be an even integer such that $2 \leq d \leq n$. Then :

$$\omega_{n+1,d+1} = 2\omega_{n,d}.$$

Sketch of proof :

$$g(x, x_{n+1}) = \begin{cases} f(x), & \text{when} \quad x_{n+1} = 0; \\ f(x + 1_n), & \text{when} \quad x_{n+1} = 1. \end{cases}$$

Constructions by product :

The Walsh transform of a direct sum equals the product of the Walsh transforms. This allows mainly to build resilient functions.

The Fourier-Hadamard transform of a direct product equals the product of the Fourier-Hadamard transforms :

$$\sum_{x\in\mathbb{F}_2^n, y\in\mathbb{F}_2^m} f(x)g(y)(-1)^{a\cdot x\oplus b\cdot y} =$$

$$\left(\sum_{x\in\mathbb{F}_2^n} f(x)(-1)^{a\cdot x}\right)\left(\sum_{y\in\mathbb{F}_2^m} g(y)(-1)^{b\cdot y}\right).$$

Multiplying Boolean functions produces unbalanced functions.

Proposition Let f_j be d_j-CI for any $1 \leq j \leq t$.
For $x^{(1)}, x^{(2)}, \ldots, x^{(t)} \in \mathbb{F}_2^n$, let :

$$h(x^{(1)}, x^{(2)}, \ldots, x^{(t)}) = \prod_{j=1}^{t} f_j\left(\sum_{i=1}^{t}\left(x^{(i)} \times M^{(i,j)}\right)\right),$$

where $M = (M^{(i,j)})_{1 \leq i,j \leq t}$ is an $nt \times nt$ nonsingular binary matrix.

M' : transposed of M^{-1}.
Assume that, if $1 \leq w_H(u^{(1)}, u^{(2)}, \ldots, u^{(t)}) \leq d$, then there exists $1 \leq j \leq t$ such that $1 \leq w_H\left(\sum_{i=1}^{t} u^{(i)} \times M'^{(i,j)}\right) \leq d_j$.

Then h is d-CI and has Hamming weight $\prod_{j=1}^{t} w_H(f_j)$.

Corollary Let $d \leq n$ and $t \geq 2$.
Let f_1 be d-CI and $f_2 \ldots, f_t$ be $\lfloor \frac{d}{2} \rfloor$-CI, and let :

$$h(x^{(1)}, x^{(2)}, \ldots, x^{(t)}) = f_1(x^{(1)}) \prod_{j=2}^{t} f_j(x^{(j)} + x^{(1)}).$$

Then h is d-CI and has Hamming weight $\prod_{j=1}^{t} w_H(f_j)$.

This implies :

$$\omega_{nt,d} \leq (\omega_{n,\lfloor \frac{d}{2} \rfloor})^{t-1} \omega_{n,d}.$$

Constructions by Kronecker sum :

The Kronecker sum of two vectors is defined as

$$(x^{(1)}, x^{(2)}) \in \mathbb{F}_2^{n_2} \times \mathbb{F}_2^{n_1} \to (x^{(1)}_{i_2} + x^{(2)}_{i_1})_{1 \le i_1 \le n_1, 1 \le i_2 \le n_2} \in \mathbb{F}_2^{n_1 n_2}.$$

Generalization : for any $I = (i_1, \cdots, i_t) \in \prod_{i=1}^{t} \{1, \ldots n_i\}$,
let us denote $I^{(r)} = (i_1, \cdots, i_{r-1}, i_{r+1}, \cdots, i_t)$, then define :

$$(x^{(1)}, x^{(2)}, \ldots, x^{(t)}) \in \mathbb{F}_2^{\sum_{i=2}^{t} n_i} \times \cdots \times \mathbb{F}_2^{\sum_{i=1}^{t-1} n_i} \to$$

$$x^{(1)} \boxplus \cdots \boxplus x^{(t)} = \Big(\sum_{r=1}^{t} x^{(r)}_{I^{(r)}} \Big)_{I \in \prod_{i=1}^{t} \{1, \ldots n_i\}} \in \mathbb{F}_2^{n_1 n_2 \cdots n_t}.$$

Proposition Assume that $2^t > d$ and :

- $f_1(x^{(1)})$ is d-CI,
- $f_2(x^{(2)})$ is $2\lfloor \frac{d}{2} \rfloor$-CI,
- for any $r = 3, 4, \ldots, t$, $f_r(x^{(r)})$ is such that, if $1 \leq w_H(v^{(r)}) \leq d$ with $w_H(v^{(r)})$ even, then $W_{f_r}(v^{(r)}) = 0$. Then h defined by :

$$\mathrm{Supp}(h) = \left\{ \left(x^{(1)} \boxplus \cdots \boxplus x^{(t)}, x^{(1)} \right) \middle| ; \right.$$

$$\left. x^{(1)} \in \mathrm{Supp}(f_1), x^{(2)} \in \mathrm{Supp}(f_2), \ldots, x^{(t)} \in \mathrm{Supp}(f_t) \right\},$$

is d-CI and has Hamming weight $\prod_{r=1}^{t} w_H(f_r)$.

In particular, if f_1 is d-CI and if f_r is $2\lfloor\frac{d}{2}\rfloor$-CI for $r = 2, \ldots, t$, then h is d-CI of Hamming weight $\prod_{r=1}^{t} w_H(f_r)$.

Corollary Let $n_1 \geq 2, n_2 \geq 3$ and let f_1 be 3-CI and f_2 2-CI. Then, h defined by :

$$\mathrm{Supp}(h) \;=\; \{\big(x^{(1)} \boxplus x^{(2)}, x^{(1)}\big) | x^{(1)} \in \mathrm{Supp}(f_1), x^{(2)} \in \mathrm{Supp}(f_2)\}.$$

is 3-CI of Hamming weight $w_H(f_1)w_H(f_2)$.

Proposition Assume $2^t > d$ and let $f_1(x^{(1)})$ be d-CI and for any $r = 2, 3, \ldots, t$, let $f_r(x^{(r)})$ be such that, for every $v^{(r)} \in \mathbb{F}_2^{n_r}$ satisfying $1 \leq w_H(v^{(r)}) \leq d$ with $w_H(v^{(r)})$ even, we have $W_{f_r}(v^{(r)}) = 0$. Then h defined by :

$$\mathrm{Supp}(h) = \{x^{(1)} \boxplus \cdots \boxplus x^{(t)} |$$
$$x^{(1)} \in \mathrm{Supp}(f_1), x^{(2)} \in \mathrm{Supp}(f_2), \ldots, x^{(t)} \in \mathrm{Supp}(f_t)\},$$

is d-CI of Hamming weight $\prod_{r=1}^{t} w_H(f_r)$.

More constructions of low-weight d-CI functions can be designed by making additional restrictions on the supports.

- Conclusion :

If people tell you that Boolean functions are of no use anymore for some domain of cryptography, do not worry !

Thank you for attention !