Correlation-immune Boolean functions and counter-measures to side channel attacks

Claude Carlet

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

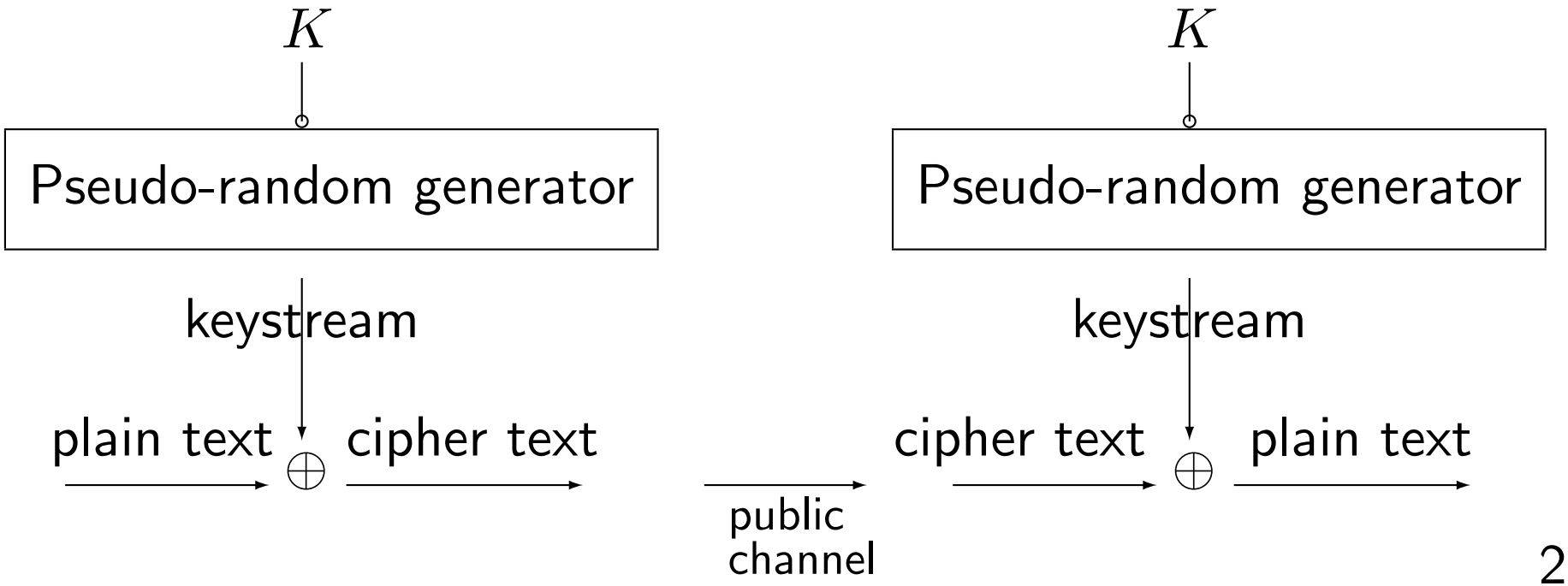
Work in common with Sylvain Guilley, Telecom Paris Tech, France

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
- New questions on correlation-immune Boolean functions

Correlation immune functions in the framework of stream ciphers

Synchronous stream ciphers :

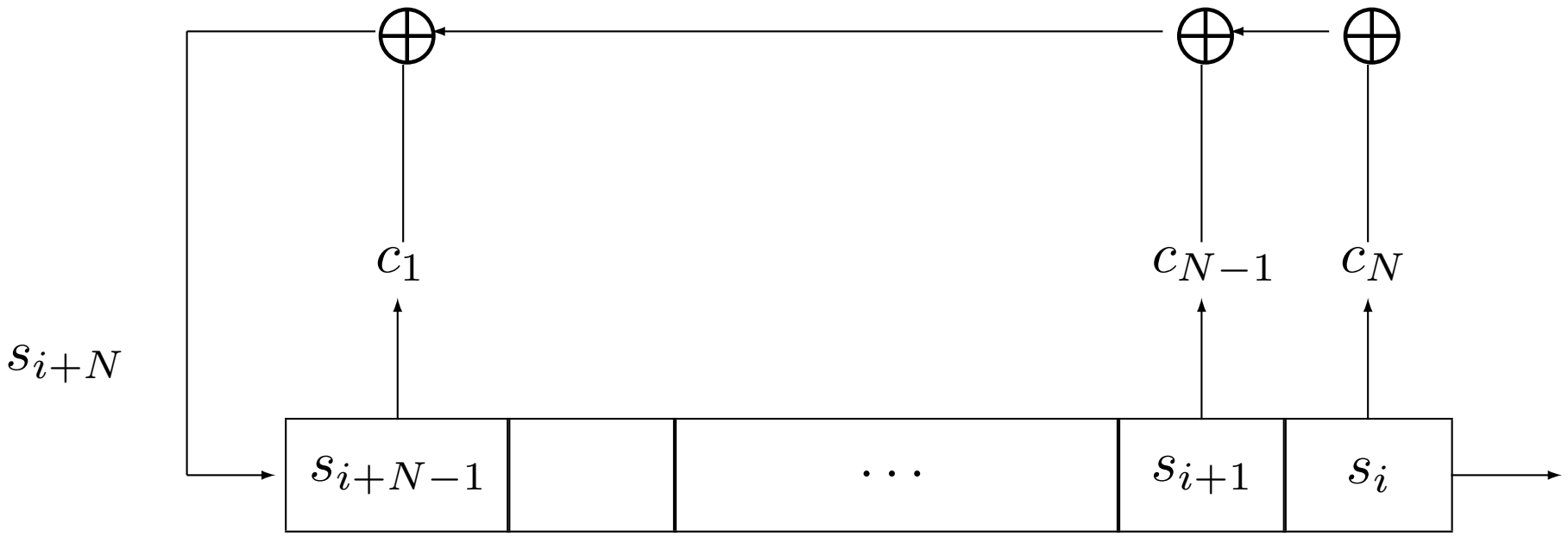


Every pseudo-random generator (PRG) consists in a linear part (for efficiency) and a nonlinear part (for robustness).

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

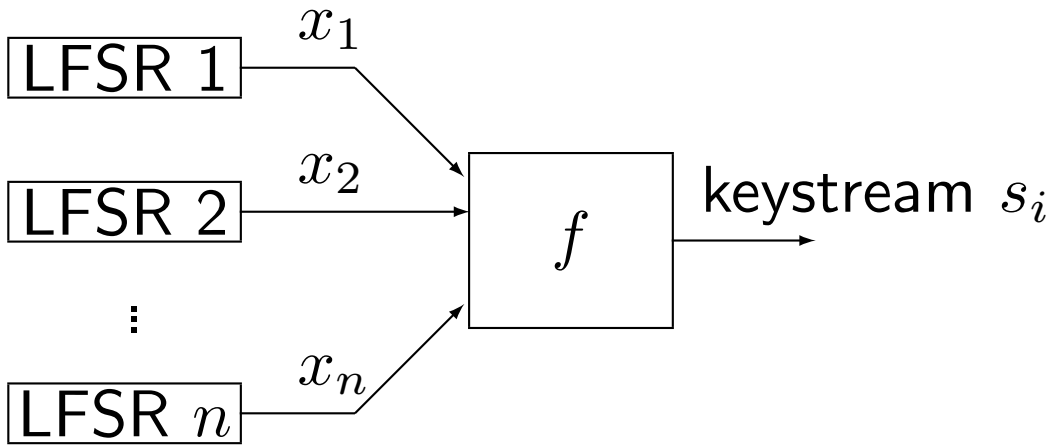
A classical **theoretical model** for their use combines the outputs of several Linear Feedback Shift Registers (LFSR) :

Linear feedback shift registers :



$$s_i = \sum_{j=1}^{N} c_j s_{i-j}.$$

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.

Equivalently, 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) :

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

where w_H is the Hamming weight :

$$w_H(a) = card\,\{i = 1, \ldots, n \,/\, a_i = 1\}.$$

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

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

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

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

Third characterization : the $|C| \times n$ array of all elements of C is an orthogonal array (with no repetition) of strength d.

Weakness of CI functions for stream ciphers :

The algebraic degree of a function is the degree of its Algebraic Normal Form (ANF)

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

Correlation immune functions have low algebraic degrees :

$$deg(f) \leq n - m.$$

They are then weak against :

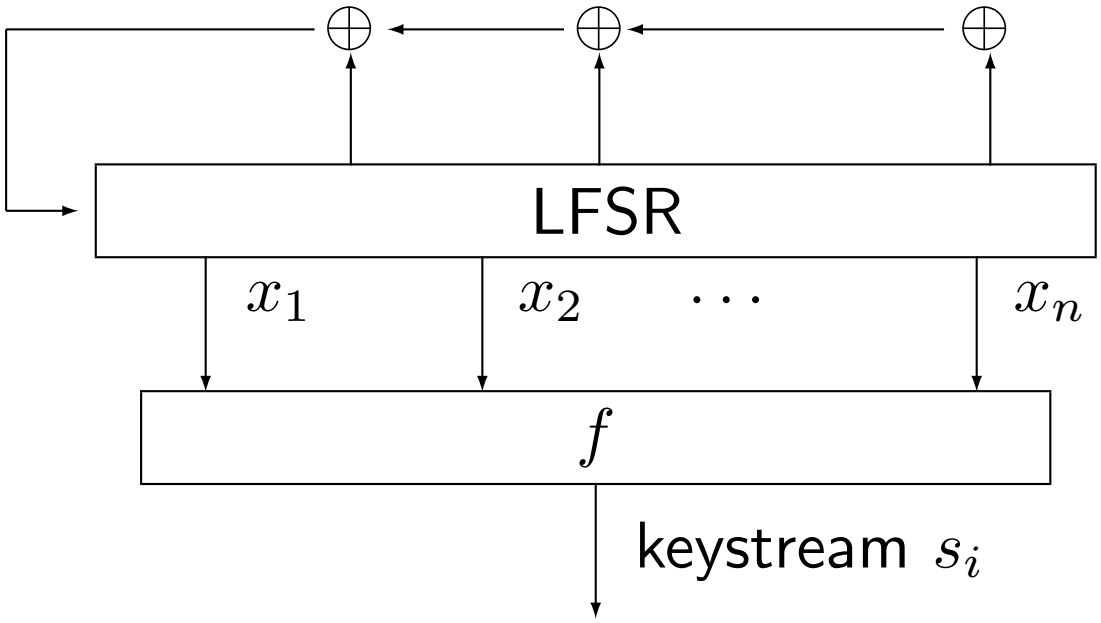
- the Berlekamp-Massey attack - complexity roughly quadratic in $L^{deg(f)}$, where L is the average size of the LFSRs,

- the Ronjom-Helleseth attack - complexity linear in $\binom{nL}{deg(f)}$,

- the fast algebraic attack, whose complexity depends on the existence of low degree functions $g \neq 0$ and h such that $fg = h$ and can be very low when f has not high algebraic degree.

Constructing functions satisfying a weakened notion of correlation immunity (C.C.-Guillot-Mesnager) and allowing resistance to all attacks is an open problem.
Consequence : another model is preferred : the filter model.

Filter model



In this model, correlation immunity is not necessary at order > 1.

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, FPGA or ASIC leaks information on the data, leading to *side channel attacks* (SCA).

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

SCA are very powerful if countermeasures are not included in the implementation of the cryptosystems, since they can use information on the data implemented inside the algorithm.

The attacker model is a *grey box* model instead of the *black box* model.

Block ciphers are particularly vulnerable to SCA because the first round (given the plaintext), or the last round (given the ciphertext) can be more easily attacked, its diffusion being not yet complete.

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

The emanations from the register are measured. They disclose a noisy version of a value related to the sensitive variable.

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

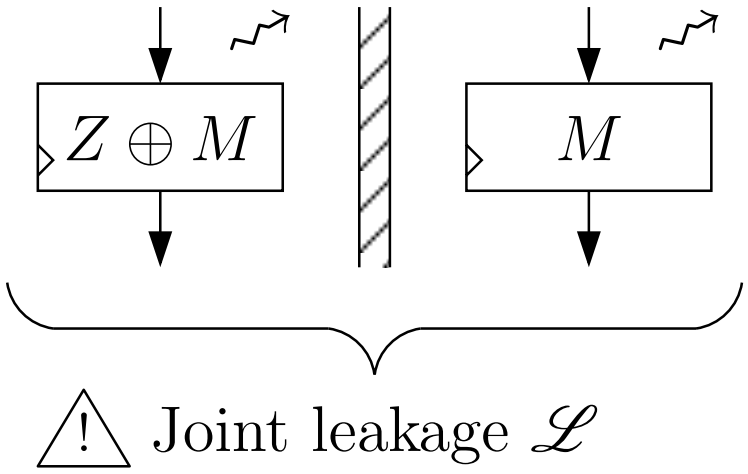
The original implementation of the AES can be attacked this way in a few seconds with a few traces.


XILINX
VIRTEX-5 FF324 PROTO

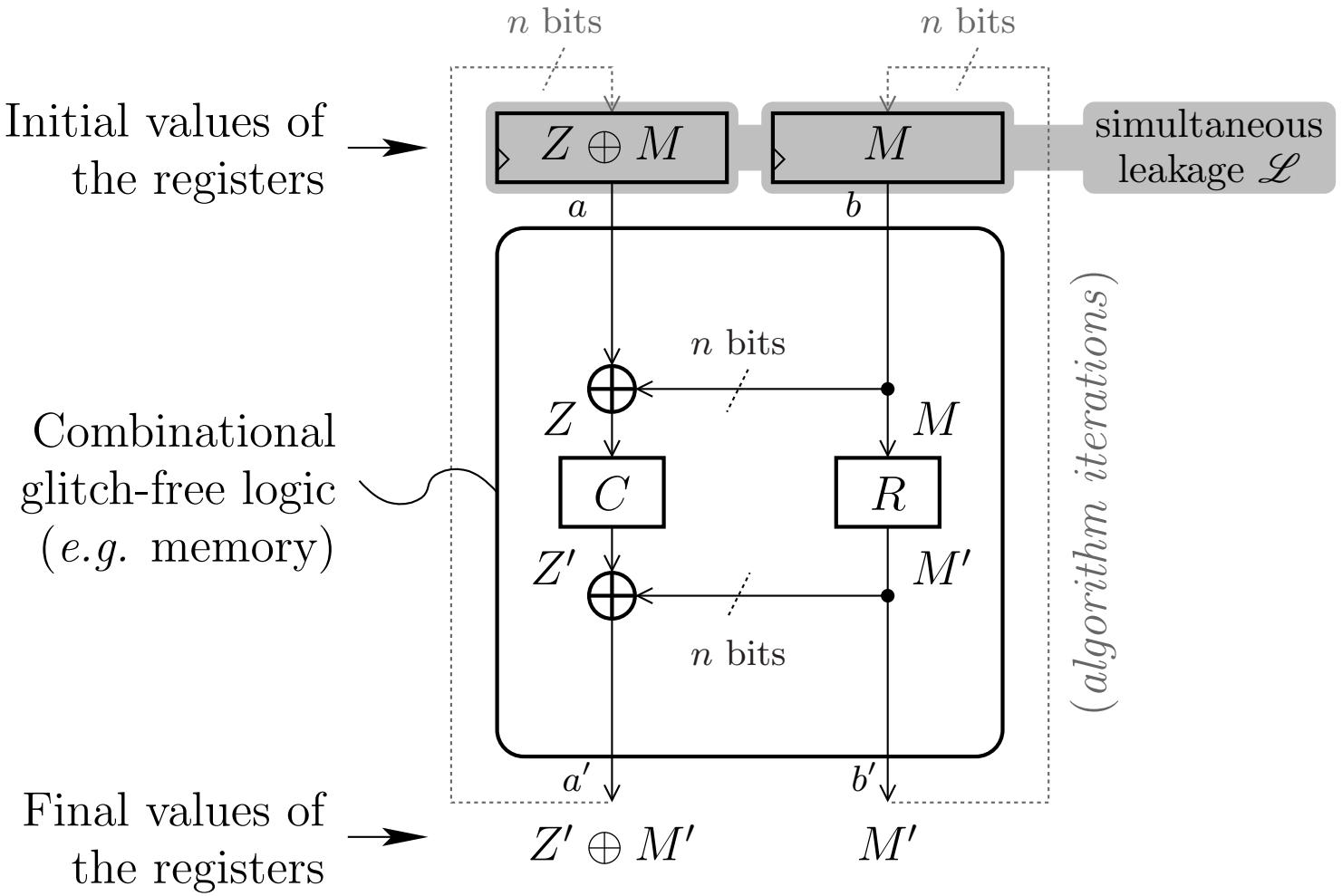
Counter-measures fortunately exist.

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 :


n bits
n bits
Initial values of the registers
$Z \oplus M$
M
simultaneous leakage $\mathscr{L}$
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
$Z' \oplus M'$
M'

This has a cost.

In software applications (smart cards), it can multiply by more than 20 the execution time when glitches are not handled (more if glitches are handled).
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 need to be modified into computations on $Z + M$ and M.

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

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

Higher order SCA consist in combining the leakages of several variables (in multivariate attacks) or, since this is often not possible, to raise the leakage at higher powers (in higher order monovariate attacks).
A second-order SCA is efficient on a single mask, but more expensive.

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

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

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

As in secret sharing, Z is hidden in $d+1$ shares M_i, such that :
– Z is a deterministic function of all the M_i, but
– Z is independent of $(M_i)_{i \in I}$ if $|I| \leqslant d$.

The cost in terms of running time and of memory is quadratic in d (cubic if the counter-measure must also deal with glitches).

The attack complexity 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).

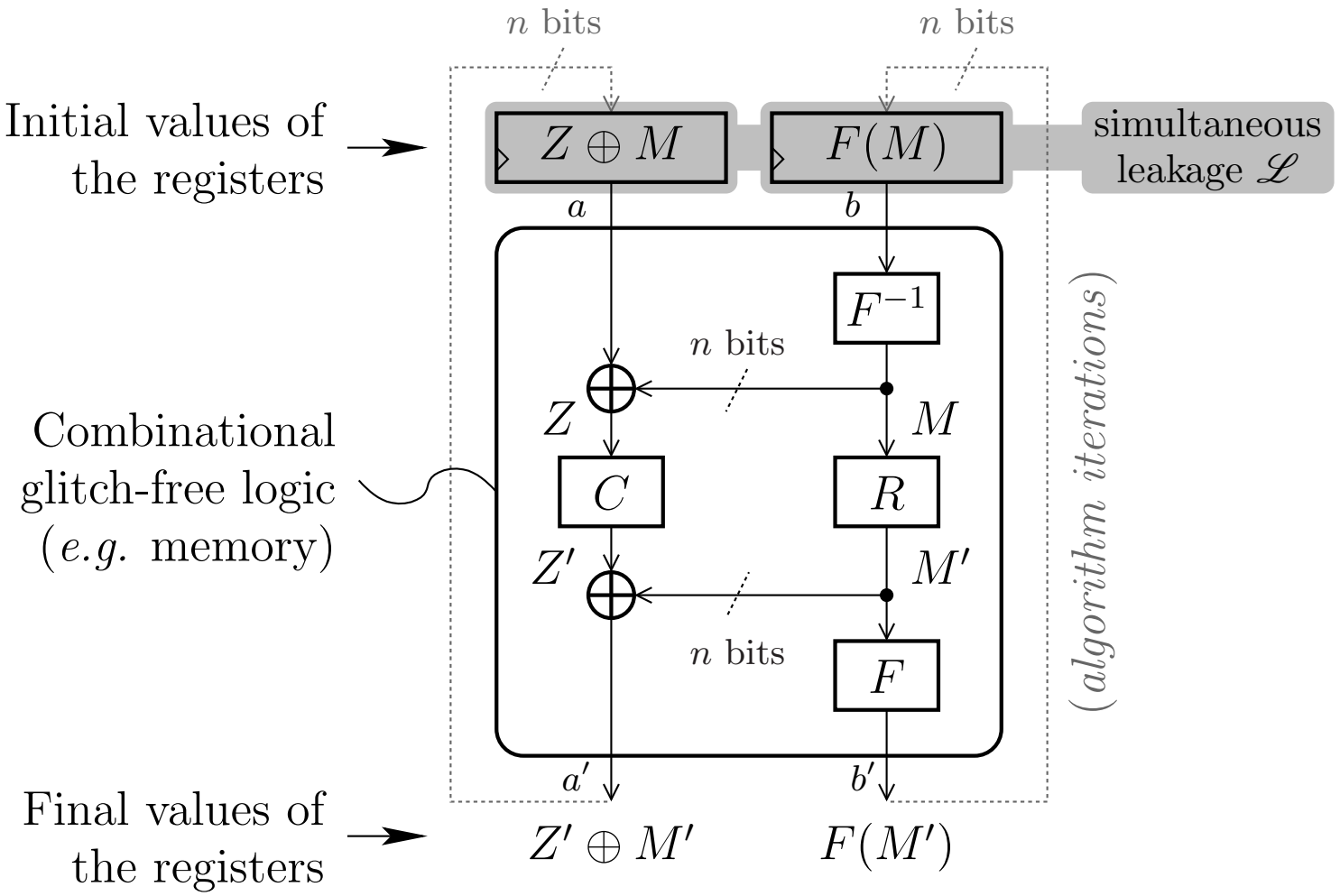
But the implementation (including masking) must be efficient today while the SCA can be performed in the future.

How Boolean functions play a new role in this framework

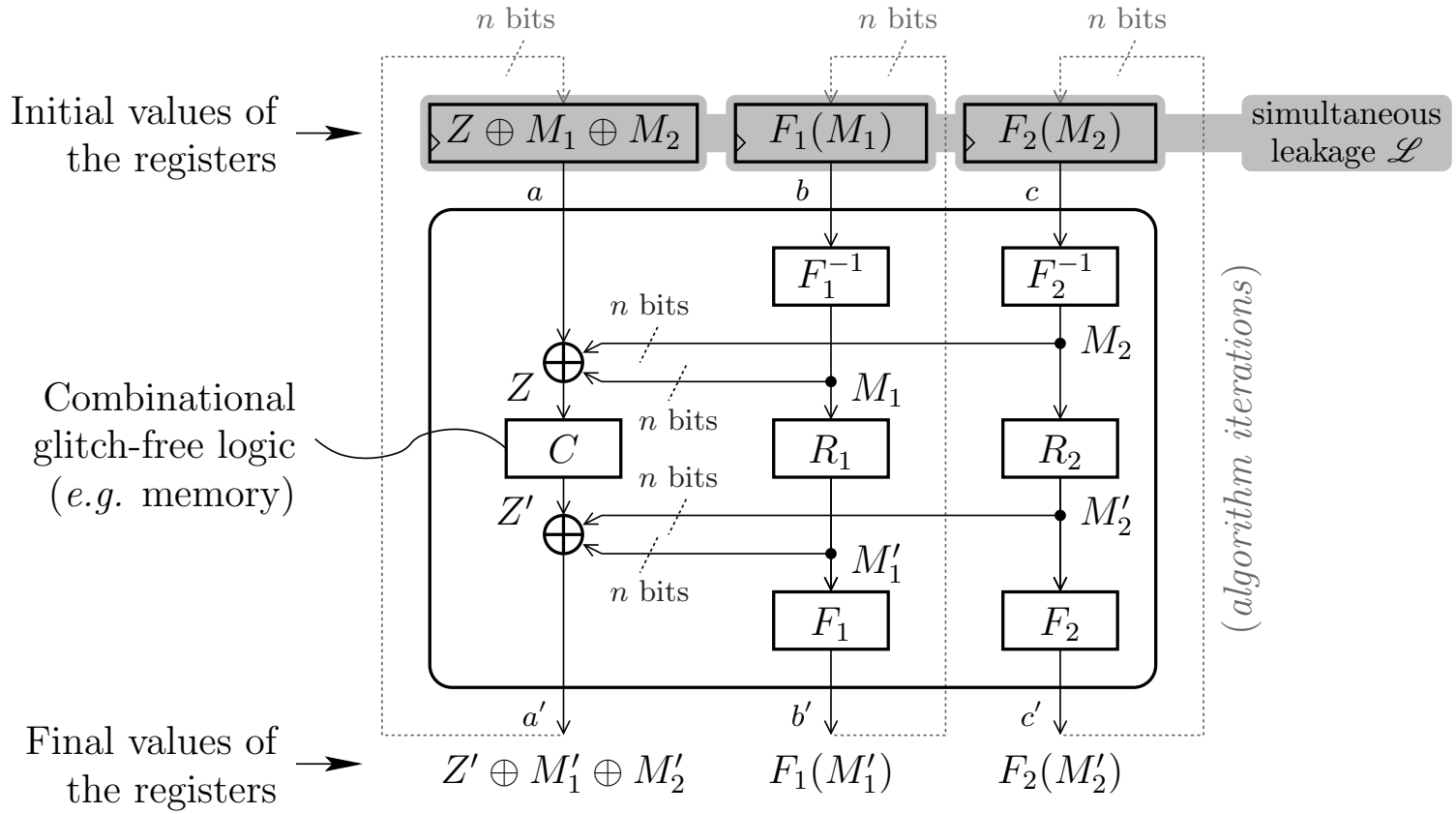
- Leakage squeezing

A setup similar to coding in digital communications, but where the goal is to make it hard for the receiver to decode the signal.

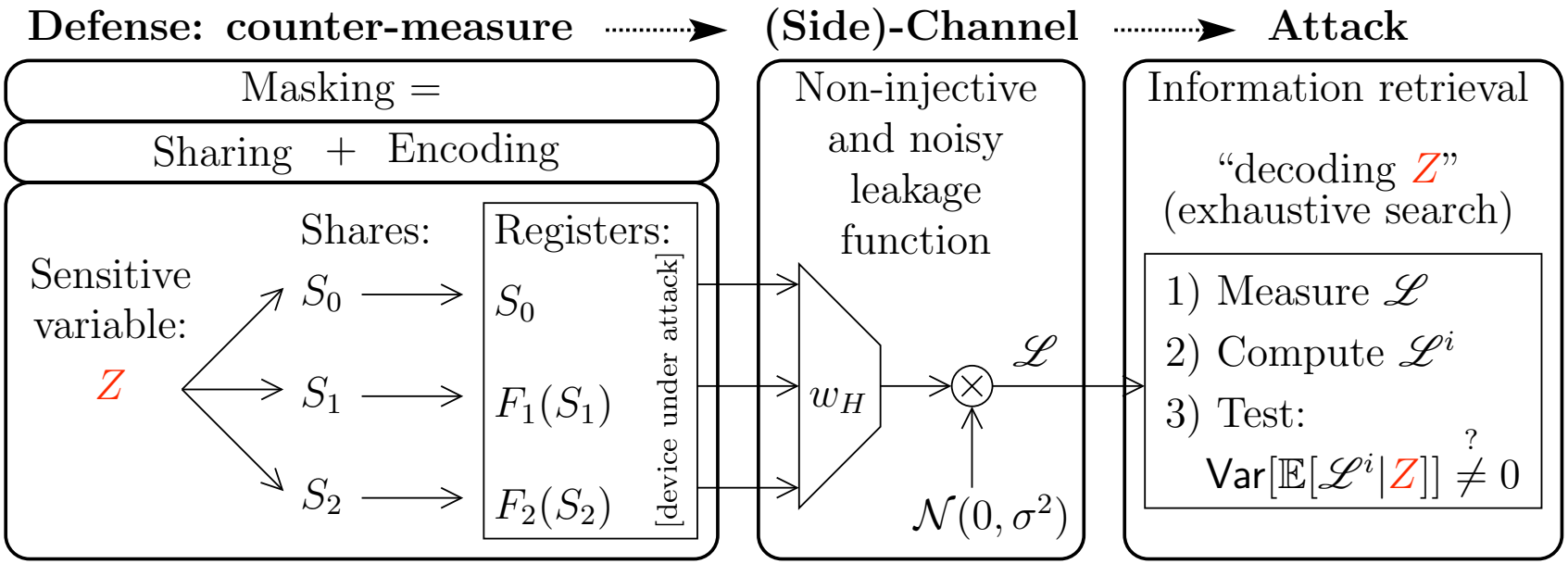
First order :


n bits
n bits
Initial values of the registers
$Z \oplus M$
$F(M)$
simultaneous leakage $\mathscr{L}$
a
b
F^{-1}
n bits
Z
M
Combinational glitch-free logic (*e.g.* memory)
C
R
Z'
M'
n bits
F
(*algorithm iterations*)
a'
b'
Final values of the registers
$Z' \oplus M'$
$F(M')$

Second order :



Attacks (on second-order leakage squeezing) :



i : *order of the attack* (increasing efficiency, increasing complexity).

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 is a set I of indices such that every possible tuple of length $|I|$ occurs in exactly one codeword within the specified coordinates $x_i;\ i \in I$.
In the case of a linear code, this means its generator matrix can have the forms $[Id_n \mid M]$ and $[N \mid Id_n]$.

Such code is called a *Complementary Information Set* (CIS) code.

There is a one-to-one correspondence between CIS codes with given information set and permutations.

The CIS codes with best dual distances have been investigated for $n \leq 65$ by C.C., P. Gaborit, J.-L. Kim, and P. Solé in the paper : *A new class of codes for Boolean masking of cryptographic computations, IEEE Trans. on Information Theory*, 2012.

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 :

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

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

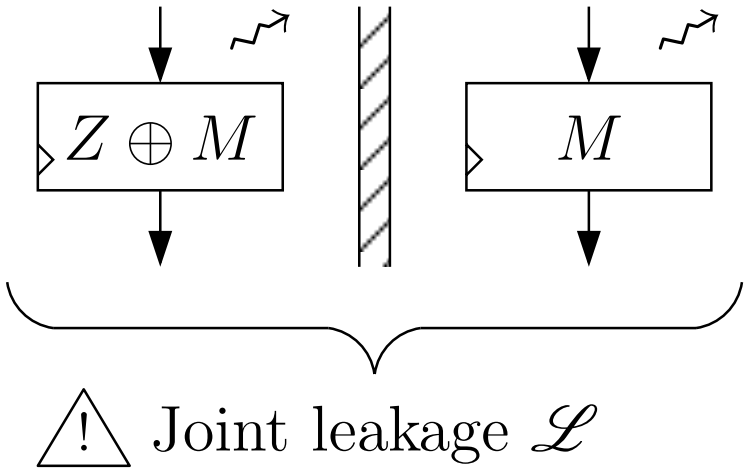
$$\sum_{x \in \mathbb{F}_2^n} (-1)^{b \cdot F_1(x) + a \cdot x} = 0 \text{ or } \sum_{x \in \mathbb{F}_2^n} (-1)^{b \cdot F_2(x) + a \cdot x} = 0.$$

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 in the paper : *Higher-order CIS codes* (submitted to IEEE Trans. on Information Theory).

- Rotating S-boxes Masking (RSM)

To avoid the joint leakage :



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

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 $d+1$.

There is no condition on this code similar to CIS, but 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.

New questions on correlation-immune Boolean functions

- What is known on CI functions :

  – Relation with orthogonal arrays (with no repetition)
  – Relation with codes (distance enumerator, dual distance)
  – Constructions :

    1. Maiorana McFarland construction :

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

2. indirect sum :

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

- What is new in our situation :

In both frameworks (leakage squeezing and RSM) the CI-functions must have low weight (and should have a particular structure in the case of leakage squeezing).

All the known constructions allow constructing balanced CI functions (called resilient) but not low weight CI-functions.

Indeed :

– $\widehat{\mathrm{f}}(a,b) = 2^r \sum_{y \in \phi^{-1}(a)} (-1)^{g(y) \oplus b \cdot y}$;
for a resilient function we can take $\phi^{-1}(0) = \emptyset$ (and then $\widehat{\mathrm{f}}(0,b) = 0$) but not for a non-balanced function ;

– $\widehat{\mathrm{h}}(a,b) = \frac{1}{2}\widehat{\mathrm{f}_1}(a)\,[\widehat{\mathrm{g}_1}(b) + \widehat{\mathrm{g}_2}(b)] + \frac{1}{2}\widehat{\mathrm{f}_2}(a)\,[\widehat{\mathrm{g}_1}(b) - \widehat{\mathrm{g}_2}(b)]$;
f_1, f_2, g_1, g_2 cannot all be balanced and there is then a problem for $a = 0$ as well (and for $b = 0$).

Challenge : find constructions of low weight CI functions.

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 $w_{n,d}$ 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.

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