Some problems on Boolean functions posed by side channel attacks

Claude Carlet

(1) LAGA, Universities of Paris 8 and Paris 13, CNRS, France

(2) University of Bergen, Norway

Outline

- Side Channel Attacks and the counter-measure of masking
- Minimizing the number of nonlinear multiplications for optimizing the masking counter-measure against side channel attacks
- Correlation immune Boolean functions for drastically reducing the cost of counter-measures against side channel attacks

Side Channel Attacks and their counter-measures

The implementation of cryptographic algorithms in devices like smart cards (mainly in software), FPGA or ASIC (in 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 grey box.

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


XILINX
VIRTEX-5 FF324 PROTO

SCA are very powerful on block ciphers if countermeasures are not included in their implementation, since SCA can use information on the data manipulated during the first round (weak diffusion).

The best classical attack on AES (most used symmetric cipher in the world) needs thousands of centuries. The original implementation of AES can be attacked by SCA in a few seconds with a few traces.

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*.

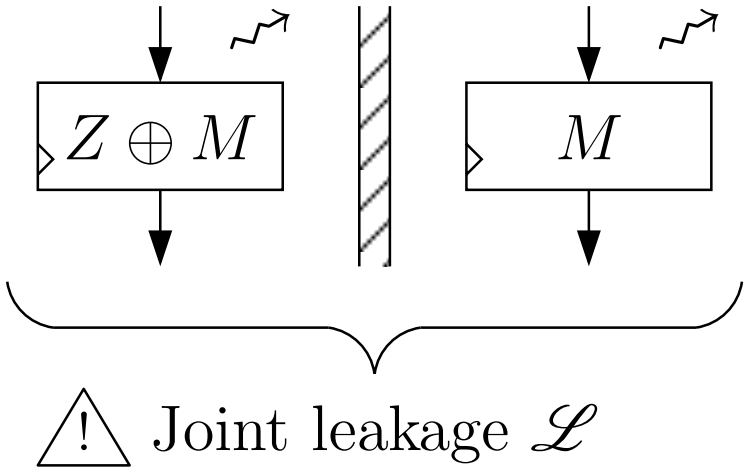
The leak discloses a noisy version of a real-valued function $\mathcal{L}$ of the sensitive variable.

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*.

Counter-measures fortunately exist.

Most common and general : *mask* each sensitive variable by splitting it : $(Z \oplus M, M)$.



If the leakage is the Hamming weight w_H, then instead of $w_H(Z)$, the attacker will have traces of $w_H(Z \oplus M) + w_H(M)$.

And the mean of $w_H(Z \oplus M) + w_H(M)$, when M is uniformly distributed, is independent of Z, since $\sum_{x \in \mathbb{F}_2^n}(w_H(z+x)+w_H(x)) = 2\sum_{x \in \mathbb{F}_2^n} w_H(x)$. But this has a cost :

- In software (smart cards), this increases the execution time.

An AES runs in less than 4000 cycles without masking and with masking it needs 100 000 cycles.

The program executable file size is also increased because all the computations need to be modified into computations on shares.

- In hardware (ASIC, FPGA), the implementation area is roughly tripled (expensive !).

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

This forces to apply higher order masking :

$d + 1$ 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 d-th order SCA* (time and number of traces) is *exponential* in the order : $O(V^d)$, where V is the variance of the noise (since taking $\mathcal{L}^d$ raises the noise at the d-th power).

The *cost* of masking (running time and memory) is *quadratic* in d.

Hence, theoretically, the designer can take advantage over the attacker but practically there is a need of reducing the cost.

Minimizing the number of nonlinear multiplications

We need, for ensuring the security of the whole algorithm, to change every function $x \mapsto F(x)$ in the algorithm into a function $(m_0, \ldots, m_d) \mapsto (m'_0, \ldots, m'_d)$ (called a *masked version of* F) such that, if $m_0, \ldots, m_d$ are shares of x then $m'_0, \ldots, m'_d$ are shares of $F(x)$.

If F is linear (like diffusion layers - MixColumns and ShiftRows in AES), then we can take $m'_i = F(m_i)$ for designing its masked version.

The case of an affine function is similar.

If F is not affine (e.g. a substitution layer - SubBytes in AES), we can consider it over the field $\mathbb{F}_{2^n}$ (always possible since $\mathbb{F}_{2^n}$ is a vector space over $\mathbb{F}_2$) and it is then representable by a univariate polynomial function, whose computation can be decomposed into a sequence of additions and multiplications in the field.

The operations of addition, scalar multiplication and squaring are linear.

For masking multiplication, there is a method called the *ISW algorithm* (Ishai-Sahai-Wagner).

The time complexity and the amount of random data which needs to be generated for the ISW algorithm are both quadratic in d.

We need to minimize the number of nonlinear multiplications.

- When the S-box is a power function $F(x) = x^d$ like in the AES, minimizing the number of nonlinear multiplications results in a variant of the classical problem of minimizing addition chains in a group.

For instance, the *inverse function* $x \to x^{254} = x^{-1}$ in $\mathbb{F}_{2^8}$ can be implemented with 4 nonlinear multiplications, in many ways (not with the square-and-multiply algorithm, though).

- When the S-box is a general polynomial, minimizing the number of nonlinear multiplications is a new paradigm. It is proved by Coron et al. that, for every positive integer n, there exists a polynomial $P(x) \in \mathbb{F}_{2^n}[x]$ with masking complexity $\mathcal{MC}(P) \geq \sqrt{\frac{2^n}{n}} - 2$.

There exist several methods for trying to minimize the multiplicative complexity :

— The *cyclotomic method* consists in rewriting $P(x)$ in the form :

$$P(x) = u_0 + \sum_{i=1}^{q} L_i(x^{\alpha_i}) + u_{2^n-1}x^{2^n-1} \ ,$$

where $(L_i)_{i \leqslant q}$ is a family of linear functions.

Upper bound on the masking complexity : $\sum_{\delta|(2^n-1)} \frac{\varphi(\delta)}{\mu(\delta)} - 1$,

where $\mu(m)$ denotes the multiplicative order of 2 modulo m and φ the Euler's totient function.

— The *Knuth-Eve method* is based on a recursive use of the decomposition :

$$P(x) = P_1(x^2) \oplus P_2(x^2)x$$

where $P_1(x)$ and $P_2(x)$ have degrees bounded above by $\lfloor deg(P)/2 \rfloor$.

Upper bound on the masking complexity :

$$\begin{cases} 3 \cdot 2^{(n/2)-1} - 2 & \text{if } n \text{ is even,} \\ 2^{(n+1)/2} - 2 & \text{if } n \text{ is odd.} \end{cases}$$

— The *Coron-Roy-Vivek (CRV) method* (heuristic but more efficient) starts with a union $\mathcal{C}$ of cyclotomic classes $\mathcal{C}_i$ in $\mathbb{Z}/(2^n - 1)\mathbb{Z}$, such that all power functions x^j, $j \in \mathcal{C}$, can be processed with a global small enough number of nonlinear multiplications.

This set of monomials x^j spans a subspace $\mathcal{P}$ of $\mathbb{F}_{2^n}[x]$. A polynomial $R \in \mathbb{F}_{2^n}[x_1, \cdots, x_t]$ is searched such that : $P(x) = R\left(P_1(x), \cdots, P_t(x)\right)$, where the P_i's are taken in $\mathcal{P}$. The search tries to minimize $\mathcal{MC}(R) + \mu$, where μ is the number of non-linear multiplications required to build $\mathcal{C}$.
A heuristic approach (in order to speed up the process) is :

1. Building $\mathcal{C}$ such that all the powers in P_i are in $\mathcal{C} + \mathcal{C}$,
2. Fixing $P_1(x)$, ..., $P_r(x)$ in $\mathcal{P}$ and searching for $P_{r+1}(x)$, ..., $P_{2r+1}(x)$ in $\mathcal{P}$ such that :

$$P(x) = \sum_{i=1}^{r} P_i(x) \times P_{r+i}(x) + P_{2r+1}(x) \ .$$

This results in solving a linear system of $n2^n$ Boolean equations.

The complexity is $\mathcal{O}(\sqrt{2^n/n})$ (asymptotically optimal).

— The *CPRR method* is based on another algebraic decomposition heuristic principle. It decomposes $P(x)$ by means of functions of low algebraic degree :

$$P(x) = \sum_{i=1}^{t} P_i\big(Q_i(x)\big) + \sum_{i=1}^{r} L_i\big(G_i(x)\big) + L_0(x) \ ,$$

where the P_i's have algebraic degree at most s and the Q_i's and G_i's are compositions of polynomials of algebraic degree at most s (the L_i's being linear).

Recall : The algebraic degree of a vectorial function $F(x)$; $x \in \mathbb{F}_2^n$, is the global degree of its ANF :

$$F(x) = \sum_{I \subseteq \{1,\ldots,n\}} a_I \prod_{i \in I} x_i; \quad a_I \in \mathbb{F}_2^n,$$

and when F_2^n is endowed with the structure of the field $\mathbb{F}_{2^n}$, and

$$F(x) = \sum_{j=0}^{2^n-1} a_j \, x^j; \quad a_j, \, x \in \mathbb{F}_{2^n},$$

the algebraic degree of F equals the maximal number of 1's needed for writing j in base 2 when $a_j \neq 0$.

The decomposition step starts by deriving a family of generators : $\begin{cases} G_1(x) = F_1(x) \\ G_i(x) = F_i\big(G_{i-1}(x)\big) \end{cases}$ where the F_i are random polynomials of algebraic degree s.
Then it randomly generates t polynomials $Q_i = \sum_{j=1}^{r} L_j \circ G_j$, where the L_j are linearized polynomials.
Eventually, it searches for t polynomials P_i of algebraic degree s and for $r+1$ linearized polynomials L_i such that :

$$P(x) = \sum_{i=1}^{t} P_i\big(Q_i(x)\big) + \sum_{i=1}^{r} L_i\big(G_i(x)\big) + L_0(x) \ .$$

This is solving a system of linear equations over $\mathbb{F}_{2^n}$.

For masking a function F of algebraic degree at most s, the method uses that, for every function from $\mathbb{F}_{2^n}$ to itself of algebraic degree at most s, the mapping

$$\varphi_F^{(s)}(a_1, a_2, \ldots, a_s) = \sum_{I \subseteq \{1,\ldots,s\}} F\Big(\sum_{i \in I} a_i\Big)$$

is multilinear, which allows proving that, for every $d \geq s$:

$$F\Big(\sum_{i=1}^{d} a_i\Big) = \sum_{j=0}^{s} \mu_{d,s}(j) \sum_{\substack{I \subseteq \{1,\ldots,d\} \\ |I|=j}} F\Big(\sum_{i \in I} a_i\Big),$$

where $\mu_{d,s}(j) = \binom{d-j-1}{s-j} \bmod 2$ for every $j \leq s$.

This reduces the complexity of the d-masking of a degree s function to several s-maskings
$\rightarrow$ Better complexity when the field multiplication is a costly operation (exceeds 5 elementary operations).

Remark.

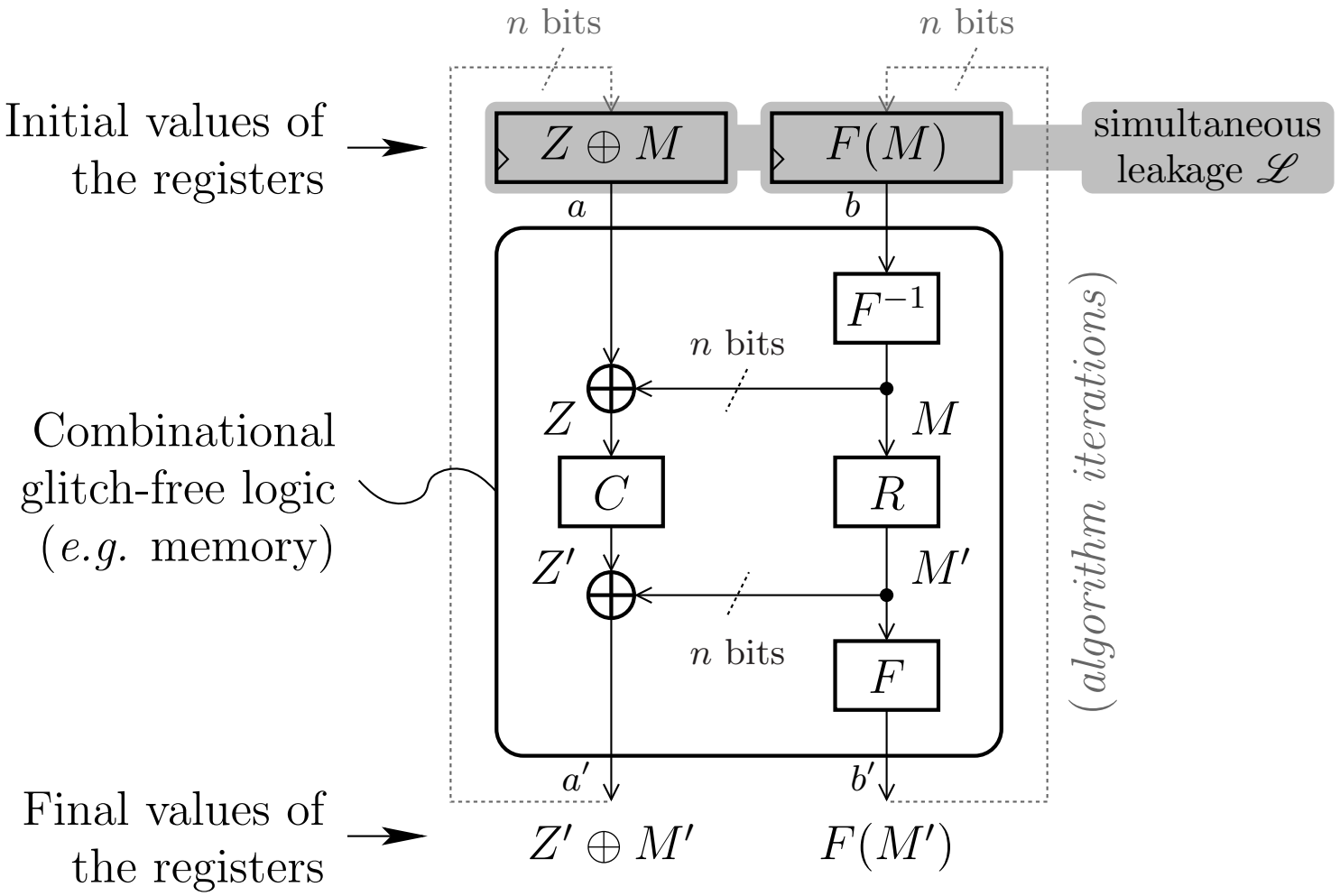
1. This property has been recently used to generate a universal *Threshold implementation* of S-boxes, which minimizes the number of shares (Piccione, Budaghyan, C.C., Nikova, Rijmen et al.).
2. Some block ciphers have been designed with S-boxes minimizing the masking complexity, such as PICARO (Piret-C.C.-Roche).

Correlation immune Boolean functions for counter-measures against side channel attacks

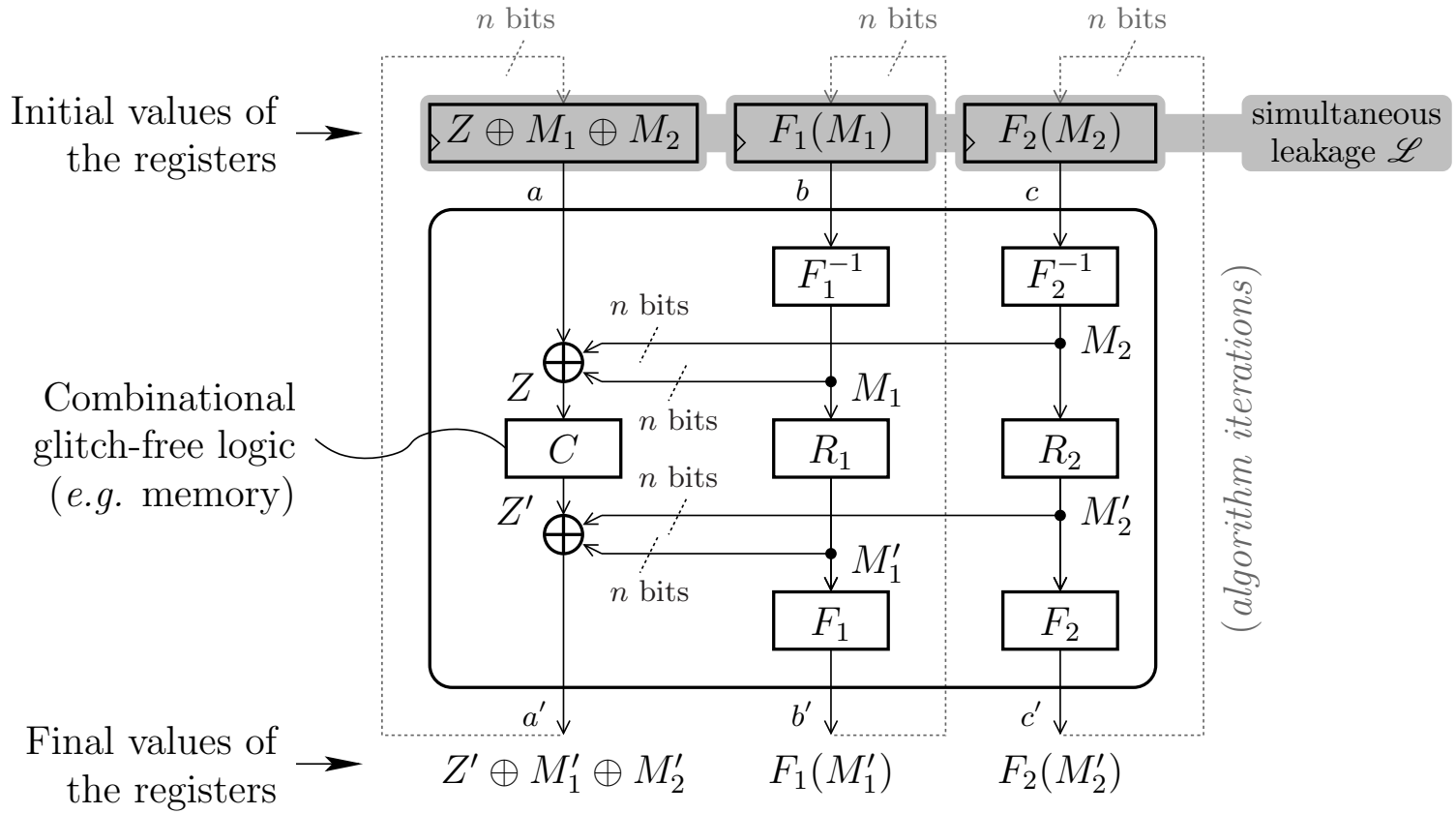
- Leakage squeezing

A masking method in hardware that is similar to coding in digital communications, but where the goal instead of allowing correction of errors, is to make it hard for the attacker to decode the signal.

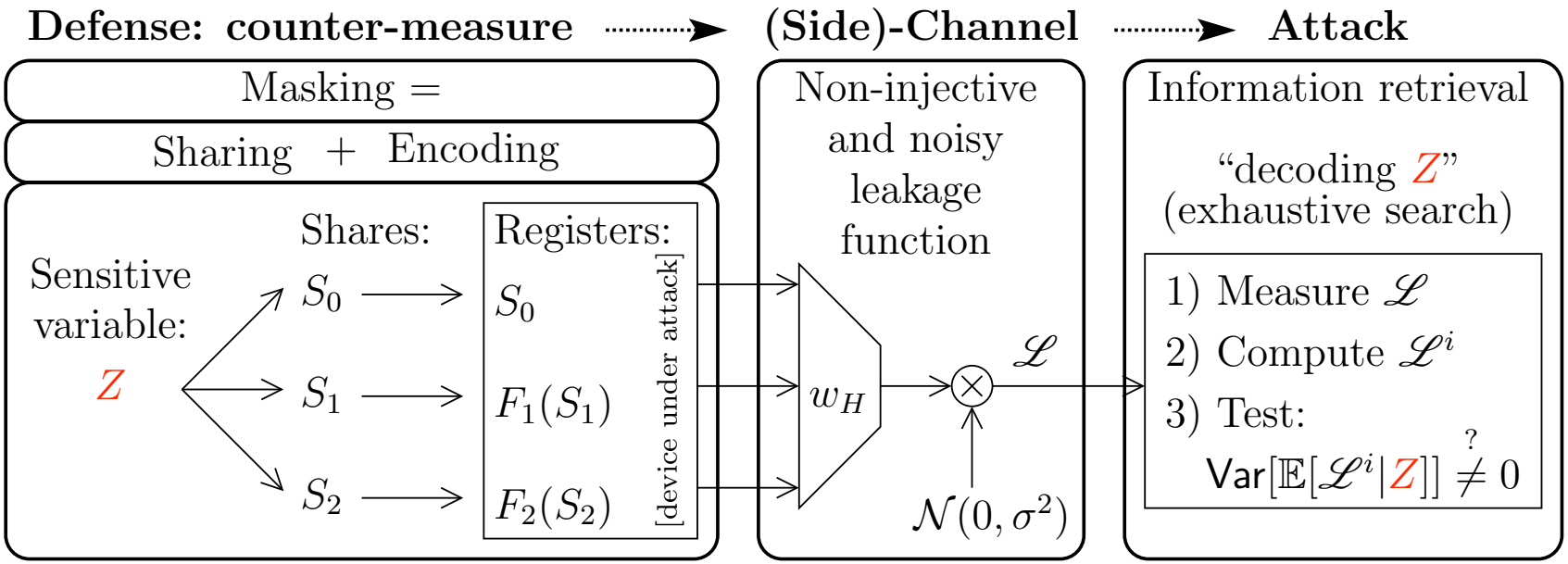
First order :


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) :



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

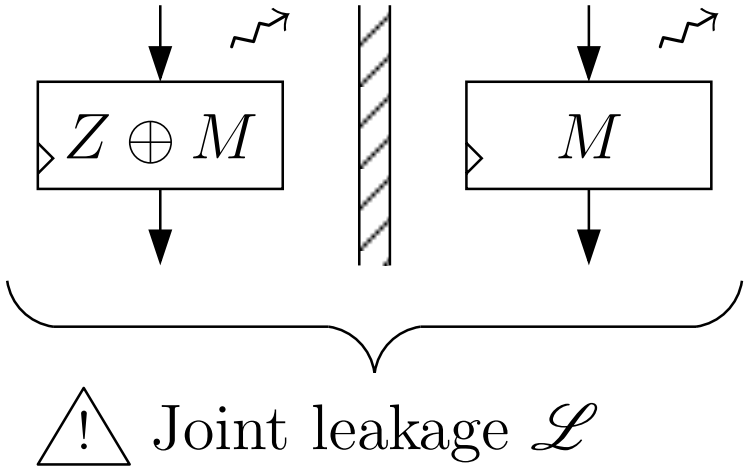
$$\mathcal{G}_F = \{(x, F(x), x \in \mathbb{F}_2^n\}$$

of F is a *d-th order correlation immune* (d-CI) function.

Equivalently, the systematic code $\mathcal{G}_F$ has dual distance at least $d+1$.

- 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 $S'(x) = S(x \oplus M) \oplus M'$.

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 AES) are then embedded already masked in the implementation.

Theorem [C.C., S. Guilley] The countermeasure resists the d-th order attack if and only if the indicator f of the mask set satisfies

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

Recall that such function is called d-CI (correlation immune).

Equivalently, the mask set is a code of *dual distance* at least $d+1$ (a parameter in coding theory).

We look for such functions of *minimum nonzero Hamming weight*.

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

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

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

$$d \text{ even }, n \geq d \geq 2; \omega_{n+1,d+1} = 2\omega_{n,d}.$$

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	?	256	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* $w_H(f)$*, where* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *is* d*-CI.*

Conjecture : the columns are non-decreasing.

CI functions are a particular case of *orthogonal arrays* (OA), introduced by C.R. Rao in 1947 and used in the statistical design of experiments : writing the elements of the support of f as the rows of an $N \times n$ array A, every $N \times d$ subarray of A contains each d-tuple exactly λ times as a row (d : *strength* of the OA).
In the case of a CI function, the OA is *simple* (there is no repetition of a same row) and d is called the strength of the OA.

In the theory of orthogonal arrays, for both simple and general orthogonal arrays, the main question is to determine the minimum value of N for each value of d.

This problem is known to be very hard, even for the smallest strength $d = 2$ (such orthogonal arrays with n columns and $n + 1$ rows are equivalent to Hadamard matrices).

Remark : Whether $n \mapsto \omega_{n,d}$ is non-decreasing is obvious for general OA since erasing columns keeps the strength unchanged.

New result

[C.C., Rebeka Kiss,Gábor P. Nagy. Simplicity conditions for binary orthogonal arrays, to appear in Designs, Codes and Cryptography]

Let A be an $OA(N, n, s, 2u)$. Define the integer :

$$M(n, s, 2u) = \sum_{j=0}^{u} \binom{n}{j} (s-1)^j.$$

1. If $N < 2\,M(n, s, 2u)$, then A is simple.

2. If $n \geq 5$, $s = 2$, $u = 2$ and $N = 2\,M(n, 2, 4) = n^2 + n + 2$, then either A is simple, or $n = 5$ and A is obtained by the juxtaposition of two identical arrays $OA(16, 5, 2, 4)$.

Consequences :

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	**128**	256	512	1024	1024	1024	1024	2048		
12	2	16	24	**128**	**256**	**768**	1024	2048	2048	2048	2048	4096	
13	2	16	32	**128**	**256**	**1 024**	**1 536**	4096	4096	4096	4096	4096	8192
		↑	↑	↑	↑								

Non-decreasing Partly

Conclusion :

Boolean functions interfere centrally with counter-measures against SCA, as well as codes (e.g. in code-based masking).

Their study is regularly necessary for solving problems about such counter-measures.

Conversely, SCA provide interesting new questions on Boolean functions, respectively, renew the interest of some classic questions.