On the importance of considering physical attacks when implementing lightweight cryptography

Alexandre Adomnicai [1,6] Benjamin Lac [2,6] Anne Canteaut [5] Jacques J.A. Fournier [3] Laurent Masson [1]
Renaud Sirdey [4] Assia Tria [2]

[1] Trusted Objects, Rousset, France

[2] CEA-Tech, Gardanne, France

[3] CEA-Leti, Grenoble, France

[4] CEA-List, Saclay, France

[5] Inria, Paris, France

[6] ENSM-SE, Gardanne, France

Lightweight Cryptography Workshop 2016

NIST, October 17-18 2016

Table of Contents

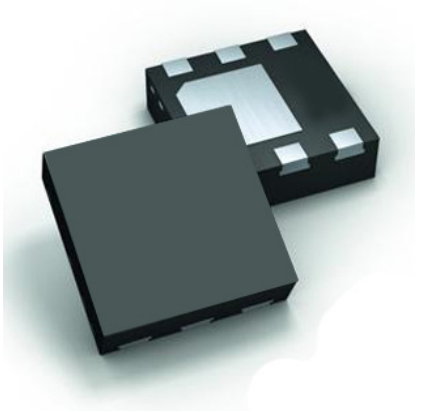
About Trusted Objects

- **Trusted Objects** is an independent company founded by experienced managers and backed up by a network of industry experts and private investors.

- **Trusted Objects**' mission is to deliver
  - **Products**: **Embedded secure firmware IPs** for IoT applications.
  - **Solutions**: **Secure Element** solution, in partnership with secure hardware provider.
  - **Services**: Security assessment & recommendations, life cycle management, **personalization**, ...

TO136 Secure Element

- A **secure element** (SE) is a tamper-resistant hardware platform, capable of **securely hosting** applications and storing confidential and cryptographic data.
- A SE can be used in addition of a host micro-controller (μC), *i.e.* the **cryptographic computations** are **delagated** to the SE via a bus, but can be also used as a **main** secure μC to handle **both application and communication**.
- The **TO136** secure element build from our **firmware** and a **secure hardware**, communicates through **I2C** bus.
- To date, our solution is made from 'traditionnal cryptography' such as
  - Elliptic Curve Cryptography (ECDSA, ECDH, ECIES, ...)
  - AES, SHA2, HMAC, ...



PRIDE block cipher 1/2

- **PRIDE** is an interative **64-bit block cipher** composed of **20 rounds** and introduced at CRYPTO 2014 by Albretch & al [1].

- We focused on PRIDE because nowadays, it is one of the most efficient lightweight block ciphers when looking at **software implementations** [2].

- As PRIDE is a simple **FX-construction** [4], it uses a **128-bit** key $k = k_0||k_1$ where k_0 is used for **pre** and **post-whitening** while k_1 is used to produce **subkeys** $f_r(k_1)$ where

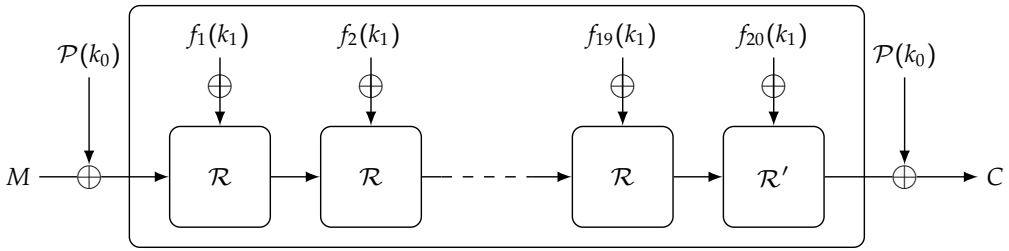
$$f_r(k_1) = k_{1_0}||g_r^{(0)}(k_{1_1})||k_{1_2}||g_r^{(1)}(k_{1_3})||k_{1_4}||g_r^{(2)}(k_{1_5})||k_{1_6}||g_r^{(3)}(k_{1_7})$$

for each round r with

$$g_r^{(i)}(x) = (x + C_i r) \bmod 256 \quad \text{and } C_i \text{ are constants.}$$

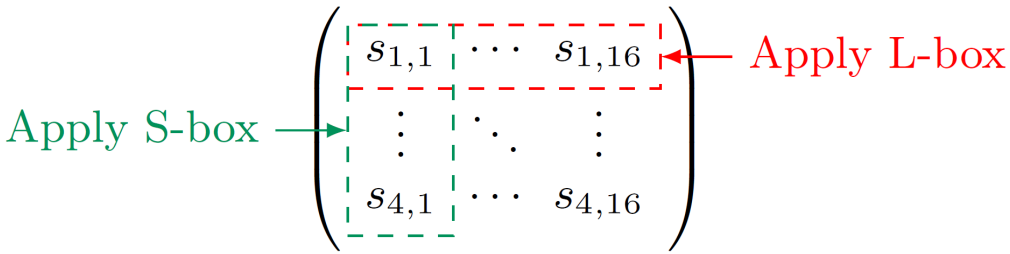
PRIDE block cipher 2/2

- Our implementation can be outlined as follows



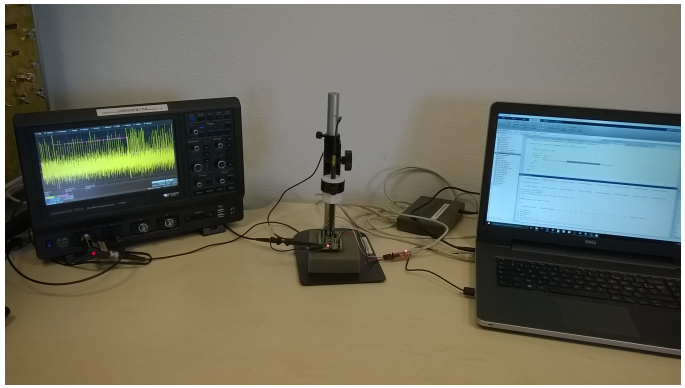
with $\mathcal{R} = \mathcal{L}-\text{layer} \circ \mathcal{S}-\text{layer}$ and $\mathcal{R}' = \mathcal{S}-\text{layer}$ where $\mathcal{S}-\text{layer} = \mathcal{P} \circ \mathcal{S} \circ \mathcal{P}^{-1}$.

- The design of PRIDE is close to **LS-design** ciphers. Each round consists in a **round key addition**, a **S-box layer** and a **L-box** one (except for the final round which omits the last operation). Hence, a round $\mathcal{R}$ can be schematized as follows



Simple Electromagnetic Analysis 1/2

- We have implemented PRIDE in **C language** on a chip embedding an **Cortex-M3** μC.
- Our attacks were performed using a **fixed key** $k = k_0||k_1$ where k_0 = `0xa371b246f90cf582` and k_1 = `0xe417d148e239ca5d`.
- A **simple electromagnetic analysis** (SEMA) on the whole execution of PRIDE was first performed in order to identify our **attack targets**.



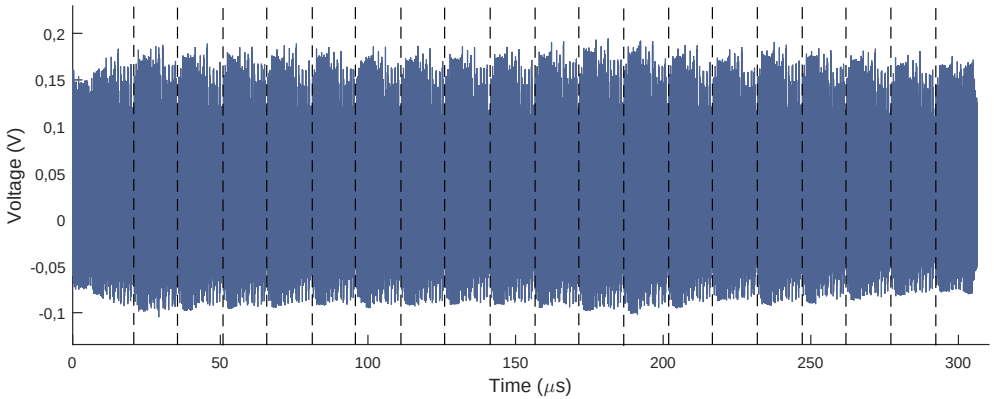


Figure: Electromagnetic emanations during a PRIDE execution

Simple Electromagnetic Analysis 2/2

- At first, it was not obvious to distinguish each **operation** within a round.
- Then, we took a look at the last round, which allowed us to determine the **different paterns** due to the **absence** of the $\mathcal{L}-$layer.

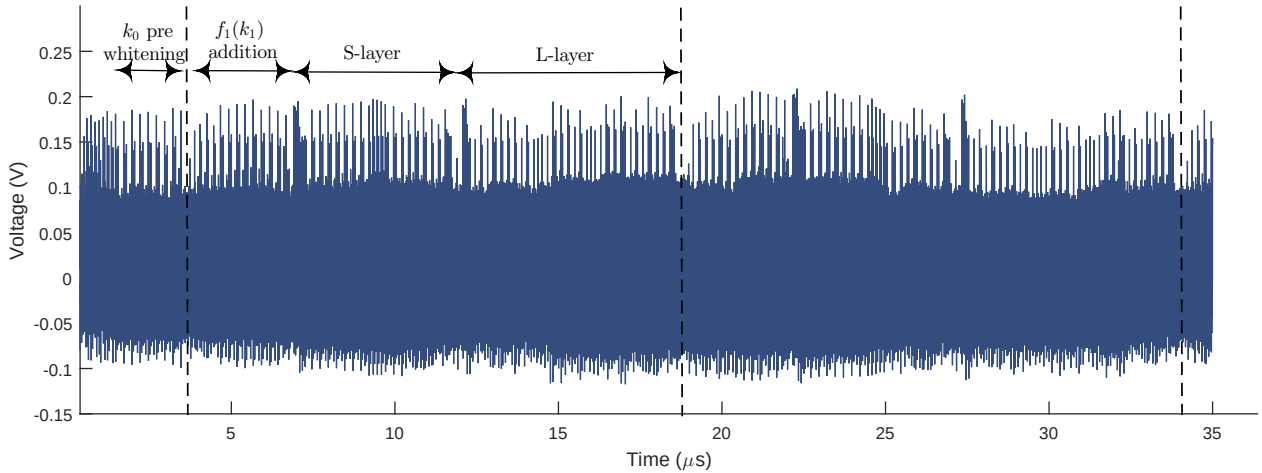


Figure: Electromagnetic emanations of the first two rounds of PRIDE block cipher

Correlation Electromagnetic Analysis

General principle

- The principle is to make the attack in two stages
  - recovering $\mathcal{P}(k_0)$
  - recovering $f_{20}(k_1)$
- We chose to focus on the **last round** because in the first one, $\mathcal{P}(k_0)$ and $f_{20}(k_1)$ are added **successively** to the state.
- The **leakage model** was based on the **Hamming weight** (HW) of the manipulated data.
- In the case of PRIDE, contrary to some other block ciphers such as AES where each byte passes through the S-box **independently**, **each byte depends on several others** during the $\mathcal{S}-$layer operation.
- We chose to attack the **key adition layer** where each byte could be treated **independently**.

$$A = c \oplus (a\&b)$$
$$B = d \oplus (b\&c)$$
$$C = a \oplus (A\&B)$$
$$D = b \oplus (B\&C)$$

PRIDE S-Box formulation on a nibble $a||b||c||d$

Correlation Electromagnetic Analysis

Experimentation

- PRIDE was executed for 1000 **random plaintexts**. The traces matrix is denoted

$$T = \begin{bmatrix} T_0 \\ \vdots \\ T_{6499} \end{bmatrix} = \begin{bmatrix} t_{0,0} & \cdots & t_{0,999} \\ \vdots & \ddots & \vdots \\ t_{6499,1} & \cdots & t_{6499,999} \end{bmatrix}.$$

- Then, we computed the **estimation matrices** in order to recover each byte $\mathcal{P}(k_0)_i$ for $0 \leq i \leq 7$

$$E^i = \begin{bmatrix} E^i_0 \\ \vdots \\ E^i_{255} \end{bmatrix} = \begin{bmatrix} e^i_{0,0} & \cdots & e^i_{0,999} \\ \vdots & \ddots & \vdots \\ e^i_{255,0} & \cdots & e^i_{255,999} \end{bmatrix}$$

where $e^i_{H_K,j} = HW(C_{j,i} \oplus H_K)$.

- Finally, we computed the **correlation coefficients** matrices P^i from E^i and T' where $T' \subset T$ denotes the traces points corresponding to the last $\mathcal{S}-$layer.

$$P^i = \begin{bmatrix} P^i_0 \\ \vdots \\ P^i_{n-1} \end{bmatrix} = \begin{bmatrix} \rho^i_{0,0} & \cdots & \rho^i_{0,255} \\ \vdots & \ddots & \vdots \\ \rho^i_{n-1,0} & \cdots & \rho^i_{n-1,255} \end{bmatrix}$$

where $\rho^i_{t,H_K} = \mathsf{Corr}(T'_t, E^i_{H_K})$.

Correlation Electromagnetic Analysis

Experimentation

- A **symmetry** about the x-axis appears because the key hypotheses are **simply XORed** with the ciphertexts.
- The **two's complement** $\overline{H_K}$ of each key byte hypothesis H_K leads to a **symmetric relation** regarding the estimation matrix $\left(i.e.\ \forall i\, \forall j,\ E^i_{\overline{H_K},j} = 8 - E^i_{H_K,j}\right)$.
- We can differentiate 8 **correlation classes** where each one corresponds to a set of key byte hypotheses $\mathcal{S}_d$ where the **Hamming distance** between the real key byte and each element equals d $(i.e.\ \forall H_K \in \mathcal{S}_d,\ HD(H_K, K) = d)$.

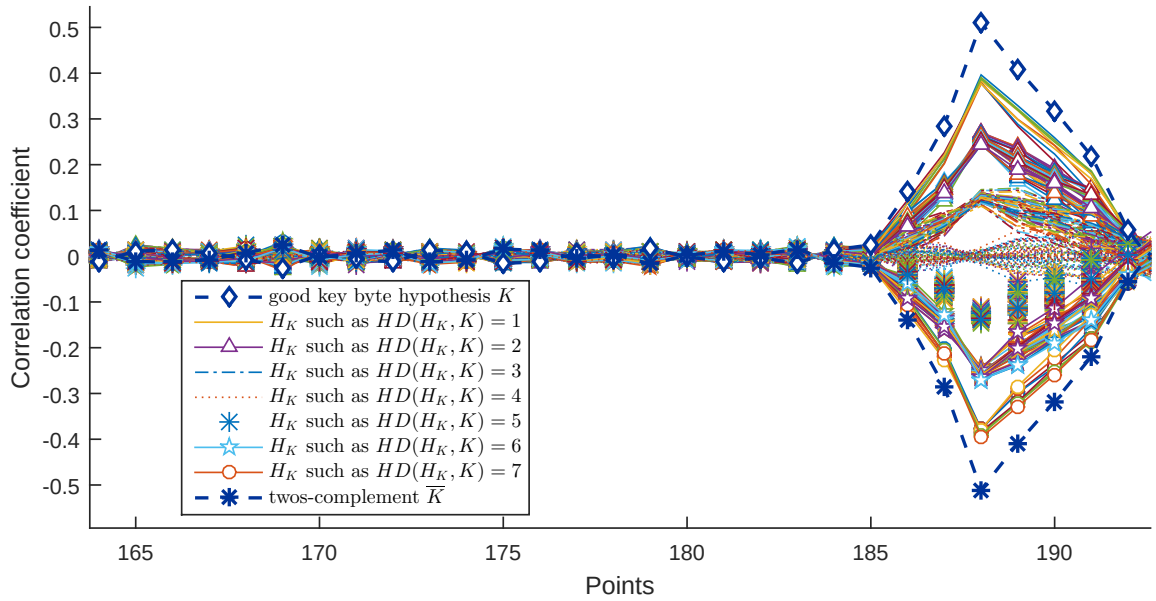


Figure: Key recovery of $\mathcal{P}(k_0)_0$ with 256-bit key hypotheses

Correlation Electromagnetic Analysis

Experimentation

- We deduced that it was sufficient to make key byte hypotheses on **7 bits** instead of 8.
- If $\max(|P^i|) = \max(P^i)$ then the correct key byte is the matching H_K, otherwise it is $\overline{H_K}$.
- In the same way, we were able to recover **all the other bytes** of $\mathcal{P}(k_0)$.
- After that, we were able to compute $\mathcal{S}-\mathsf{layer}\big(C \oplus \mathcal{P}(k_0)\big)$ for each ciphertext C and to repeat the **same reasoning** to recover $f_{20}(k_1)$.

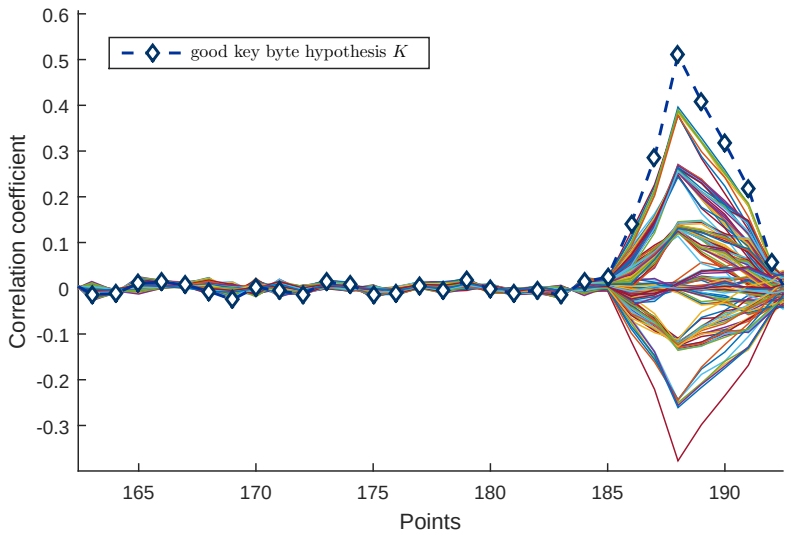


Figure: Key recovery of $\mathcal{P}(k_0)_0$ with 128-bit key hypotheses

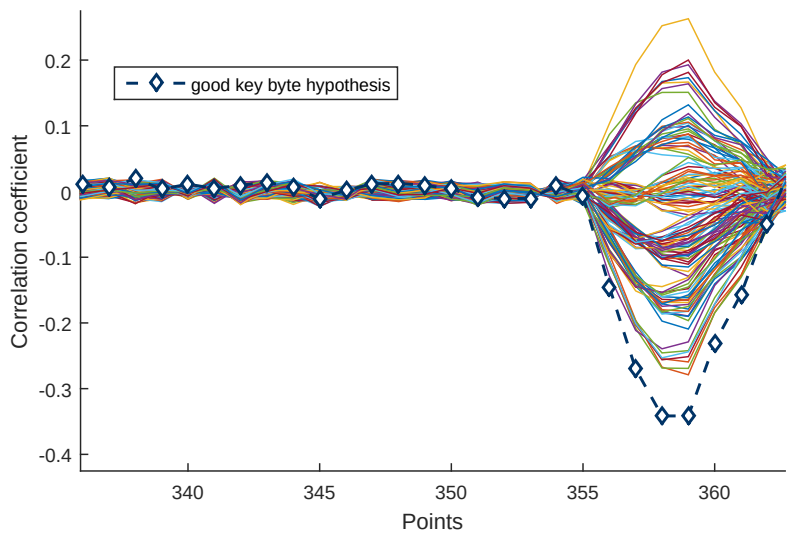


Figure: Key recovery of $\mathcal{P}(k_0)_1$ with 128-bit key hypotheses

Differential fault analysis

General principle

- We applied the attack presented in [5] on our **8-bit implementation**.
- As CEMA, the DFA consists in **two steps**.
- To recover k_0, we injected faults on some **rows** of the inner state (independently) between the **last two** $\mathcal{S}-$layer.
- A **bit flip** on the row $1 \leq \beta \leq 4$ just before the r-th $\mathcal{S}-$layer gives a **S-box input difference** $\Delta In_r = 2^{4-\beta}$.
- The **S-box output difference** can be easily recovered from the correct ciphertext C and the faulty one C^* by computing $\Delta Out_{20} = \mathcal{P}^{-1}(C \oplus C^*)$.
- We then exploited the **couples** $(\Delta In_{20}, \Delta Out_{20})$ by using the following proposition introduced in [5]

Proposition
Let $\mathcal{S}$ be an n-bit S-box with differential uniformity 4. Let (a_1, b_1) and (a_2, b_2) be two differentials with $a_1 \neq a_2$ such that the system of two equations

$$\mathcal{S}(x \oplus a_1) \oplus \mathcal{S}(x) = b_1 \quad (1)$$
$$\mathcal{S}(x \oplus a_2) \oplus \mathcal{S}(x) = b_2 \quad (2)$$

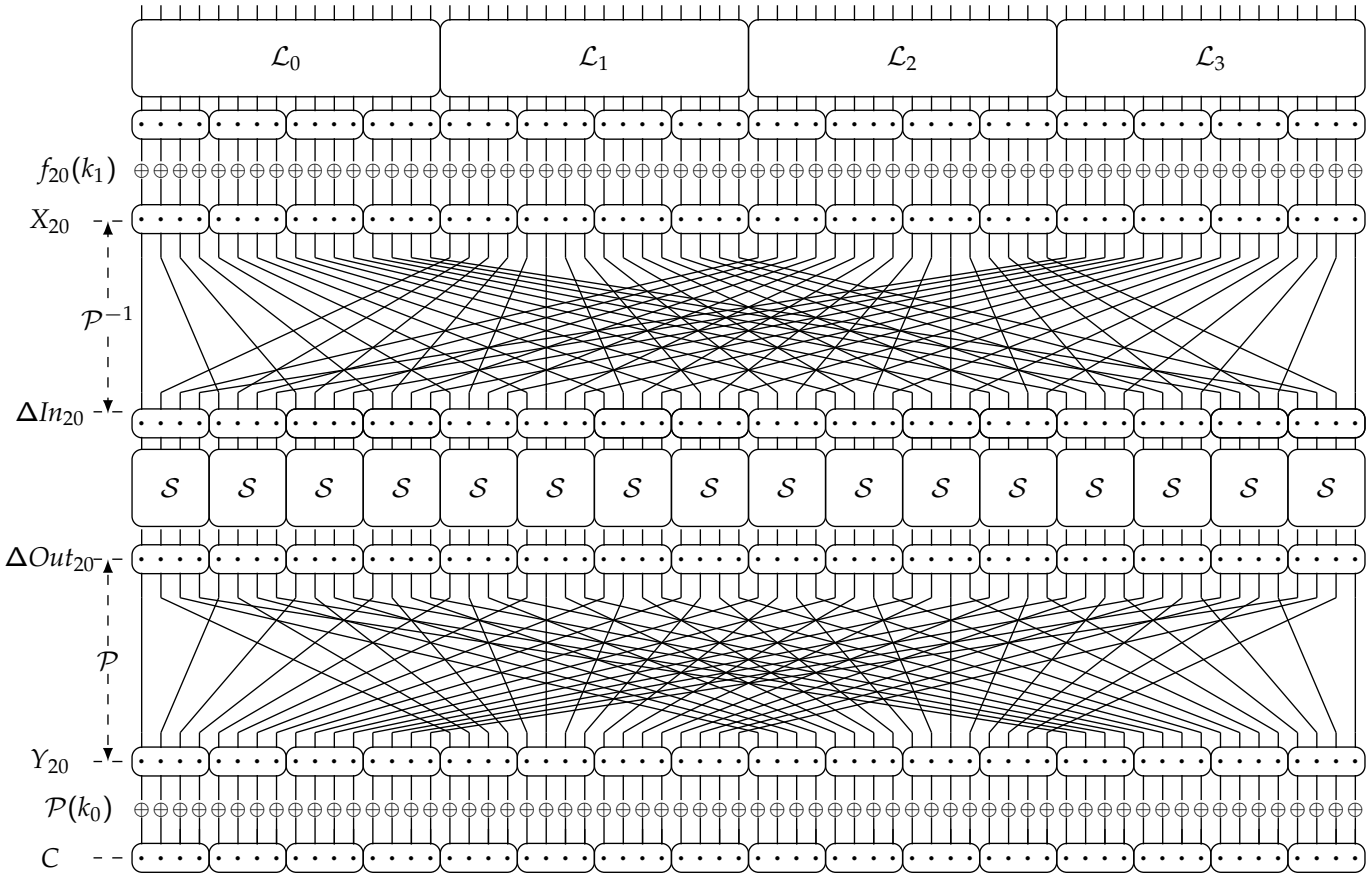
has at least two solutions. Then, each of the three equations (1), (2) and

$$\mathcal{S}(x \oplus a_1 \oplus a_2) \oplus \mathcal{S}(x) = b_1 \oplus b_2 \quad (3)$$

has at least four solutions.

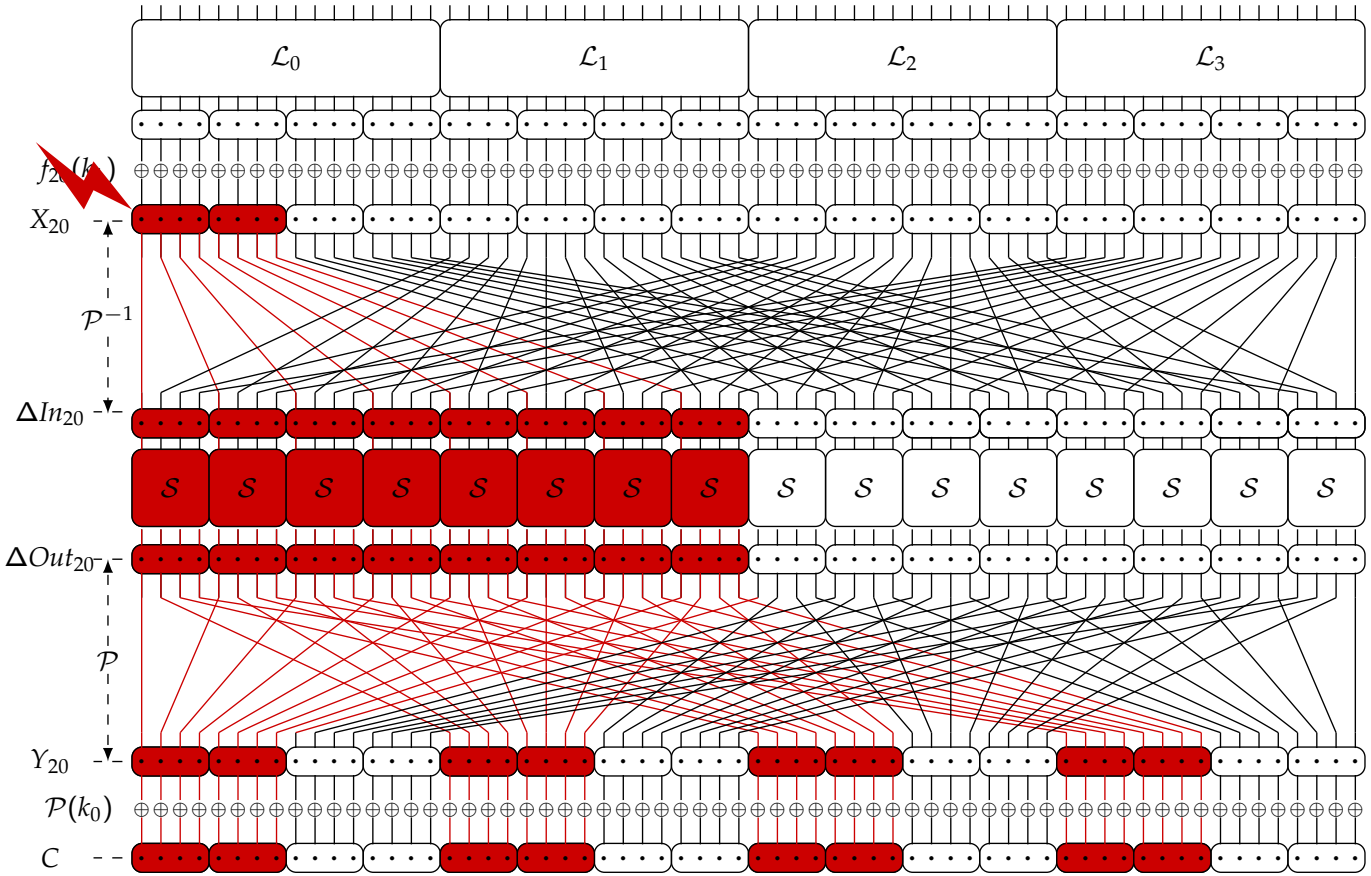
Differential Fault Analysis

Fault injection example



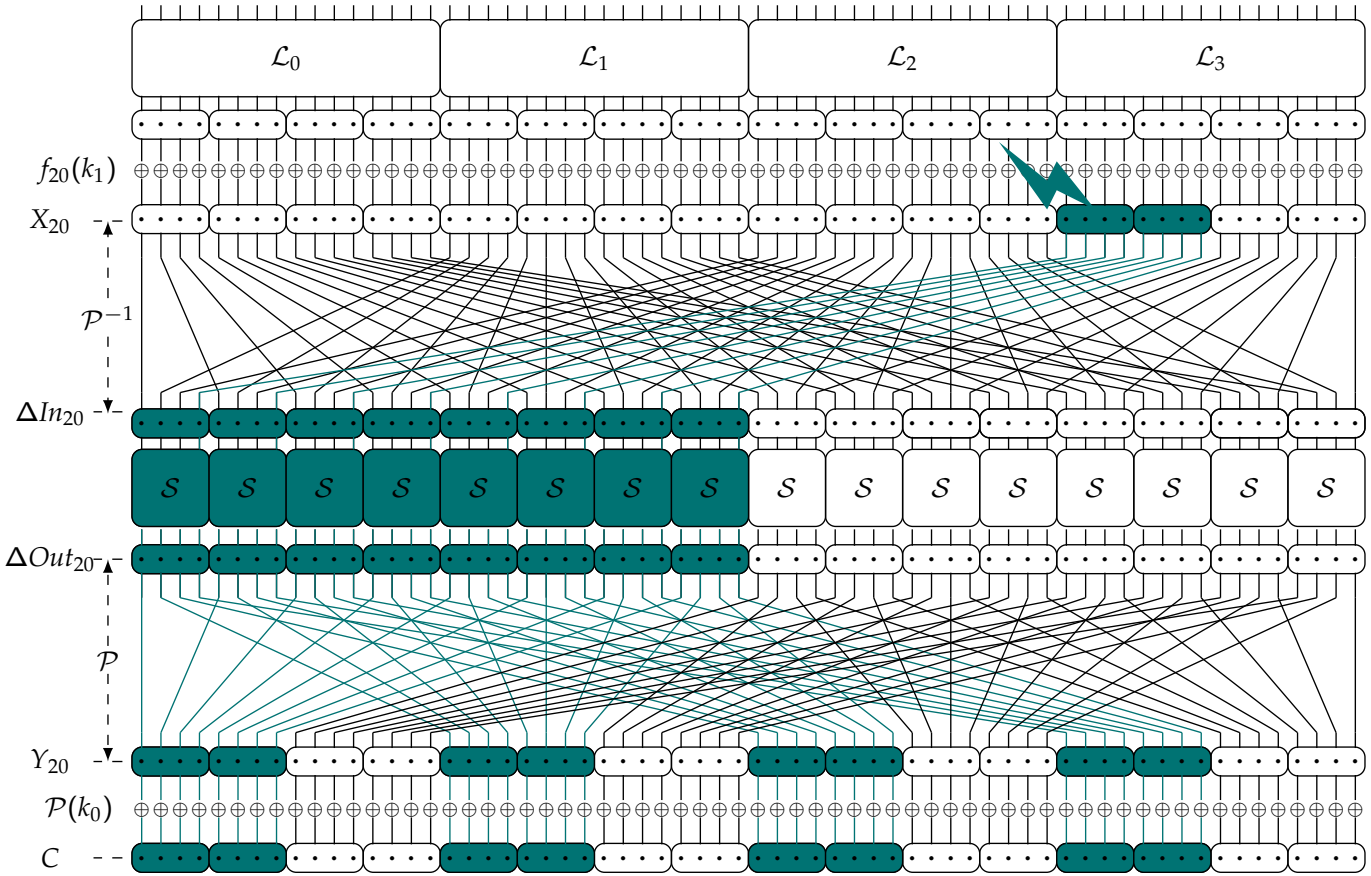
Differential Fault Analysis

Fault injection example



Differential Fault Analysis

Fault injection example



Differential fault analysis

Experimentation

Table: Sets of candidates obtained from faults injected between the last two substitution layers

Value of $(\Delta O_{20}, \Delta I_{20})$	Nib_0	Nib_1	Nib_2	Nib_3	Nib_4	Nib_5	Nib_6	Nib_7	Nib_8	Nib_9	Nib_{10}	Nib_{11}	Nib_{12}	Nib_{13}	Nib_{14}	Nib_{15}
(0xa000800000002000, 0x8000800000008000)	0x1 0x3 0x9 0xb	∅	∅	∅	0x5 0x6 0xd 0xe	∅	∅	∅	∅	∅	∅	∅	0x0 0x2 0x8 0xa	∅	∅	∅
(0xcc00df8800000000, 0x2200222200000000)	0x5 0x9	0x5 0x9	∅	∅	0x6 0xb	0x1 0xe	0x0 0x2 0x8 0xa	0x0 0x2 0x8 0xa	∅	∅	∅	∅	∅	∅	∅	∅
(0xcc0000000f000008, 0x2200000002000002)	0x5 0x9	0x5 0x9	∅	∅	∅	∅	∅	∅	∅	0x1 0xe	∅	∅	∅	∅	∅	0x0 0x2 0x8 0xa
(0xc0b00f8080f00bb0, 0x2020022020200220)	0x5 0x9	∅	0x4 0x7 0xc 0xf	∅	∅	0x1 0xe	0x0 0x2 0x8 0xa	∅	0x0 0x2 0x8 0xa	∅	0x1 0xe	∅	∅	0x4 0x7 0xc 0xf	0x4 0x7 0xc 0xf	∅
(0x0405040664707056, 0x0101010111101011)	∅	0x0 0x1 0x4 0x5	∅	0x2 0x3 0x6 0x7	∅	0x0 0x1 0x4 0x5	∅	0xa 0xb 0xc 0xd	0xa 0xb 0xc 0xd	0x0 0x1 0x4 0x5	0x8 0x9 0xe 0xf	∅	0x8 0x9 0xe 0xf	∅	0x2 0x3 0x6 0x7	0xa 0xb 0xc 0xd
(0x7005500660057006, 0x1001100110011001)	0x8 0x9 0xe 0xf	∅	∅	0x2 0x3 0x6 0x7	0x2 0x3 0x6 0x7	∅	∅	0xa 0xb 0xc 0xd	0xa 0xb 0xc 0xd	∅	∅	0x2 0x3 0x6 0x7	0x8 0x9 0xe 0xf	∅	∅	0xa 0xb 0xc 0xd
(0x7445546660700406, 0x1111111110100101)	0x8 0x9 0xe 0xf	0x0 0x1 0x4 0x5	0x0 0x1 0x4 0x5	0x2 0x3 0x6 0x7	0x2 0x3 0x6 0x7	0x0 0x1 0x4 0x5	0xa 0xb 0xc 0xd	0xa 0xb 0xc 0xd	0xa 0xb 0xc 0xd	∅	0x8 0x9 0xe 0xf	∅	∅	0x0 0x1 0x4 0x5	∅	0xa 0xb 0xc 0xd

- Because the faults did not provide enough information for the 3-rd and the 11-th nibble, 16 candidates remained for $\mathcal{P}(k_0)$.

Differential fault analysis

Experimentation

- Faulty ciphertexts obtained from fault injection between the **penultimate** two substitution layers allowed us to exclude the **bad assumptions** by computing

$$\Delta Out_{19} = \left(\mathcal{P}^{-1} \circ \mathcal{L} - \text{layer}^{-1}\right)\left(\mathcal{S} - \text{layer}\left(C \oplus \mathcal{P}(k_0)\right) \oplus \mathcal{S} - \text{layer}\left(C^* \oplus \mathcal{P}(k_0)\right)\right)$$

  from **all** the 16 remaining candidates.

- We observed that some differentials $(\Delta Out_{19}, \Delta In_{19})$ were not possible: each input difference implies a **specific** output difference set.

- The last remaining value was k_0 = `0xa371b246f90cf582`.

- Finally, we did the intersection between the sets for each nibble as we did for k_0 and we **directly** recovered k_1.

Costs analysis

- Practical feasibility
  - A CEMA can be easily set up as it does not necessarily require much equipment. The involved tools mainly depends on the targeted platform.
  - Fault attacks are very powerful but a little more complicated to set up. For our attack, we did not need to decapsulate the chip and an electromagnetic pulse generator and a picoscope did the job, but on secured platforms...

- Attack paths

  On one hand, the $\mathcal{S}-$layer design makes CEMA more tricky
  - To make a hypothesis on a **8-bit** value at the $\mathcal{S}-$layer output, one should make a hypothesis on **24-bit** input value.
  - Bit-per-bit SCAs would be more efficient but are more appropriate to hardware implementation. Such an attack has already been performed on PRINCE [6] which has a similar structure to PRIDE

  On the other hand, it makes DFA much easier
  - Flipping the 16 bits of any row at its input activates all S-boxes in the next round.
  - The number of remaining candidates for k_0 is upper-bounded by 4^{16}.

Countermeasures

Against CEMA

- For a nibble denoted $n = a \,||\, b \,||\, c \,||\, d$, a **mask** of first order $m = m_a \,||\, m_b \,||\, m_c \,||\, m_d$ and $\widetilde{n} = n \oplus m = \widetilde{a} \,||\, \widetilde{b} \,||\, \widetilde{c} \,||\, \widetilde{d}$, the S-Box returns the output nibble $\widetilde{N} = \widetilde{A} \,||\, \widetilde{B} \,||\, \widetilde{C} \,||\, \widetilde{D}$ where

$$\widetilde{A} = \widetilde{c} \oplus (\widetilde{a} \cdot \widetilde{b})$$
$$\widetilde{B} = \widetilde{d} \oplus (\widetilde{b} \cdot \widetilde{c})$$
$$\widetilde{C} = \widetilde{a} \oplus (\widetilde{A} \cdot \widetilde{B})$$
$$\widetilde{D} = \widetilde{b} \oplus (\widetilde{B} \cdot \widetilde{C})$$

- The **secure AND gate construction** proposed in [7] consists in introducing a random bit r and computing

$$m_z = r \tag{4}$$
$$\widetilde{z} = (\widetilde{a} \cdot \widetilde{b}) \oplus (m_a \cdot m_b) \oplus (m_a \cdot \widetilde{b}) \oplus (m_b \cdot \widetilde{a}) \oplus r$$

- In the particular case of PRIDE, we will need to **generate 4 random bits** (r_A, r_B, r_C, r_D) for **each** secure AND gate.

Countermeasures

Against DFA

- **Duplicating** the last rounds computations is a simple countermeasure against fault attacks.
- If computations return **different results**, it means that a fault has been injected and that the device must **react** to it.
- We can also apply a **majority vote** by duplicating the computations twice and give as output the one that **appears most**.

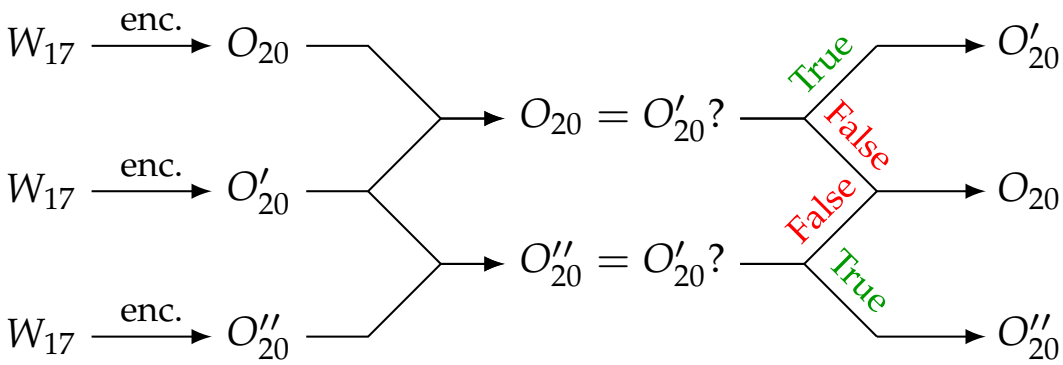


Figure: Majority vote using duplication

Countermeasures

Against both

- A countermeasure proposed in [3] consists in adding a **random mask** to the message in order to prevent **consecutive executions** of the same plaintext.
- The mask can be **sent with the ciphertext** but does not protect against an **attack on decryption**: an attacker can choose the **same mask**.
- Another option is to **synchronize** PRNGs.

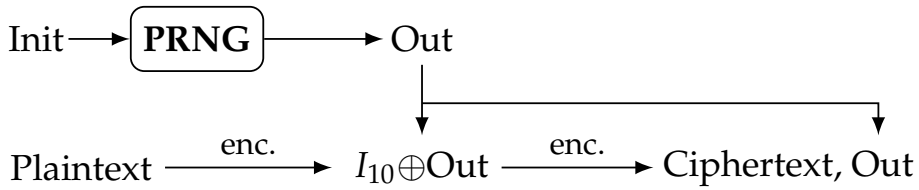
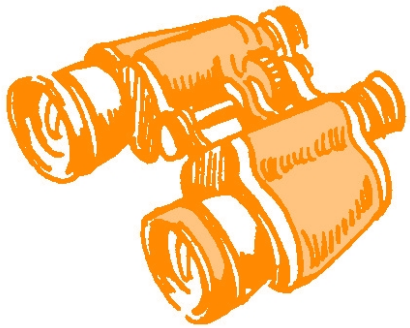


Figure: Masking based on the Guilley countermeasure

Conclusion & Perspectives

- We showed that PRIDE is **vulnerable** to CEMA as well as DFA and compared the attacks to the $\mathcal{S}-$layer design.
- A cryptographic algorithm can be **intrinsically** more resistant to physical attacks thanks to its **design**.
- Now, the next step shall be to analyse the **countermeasures' effects** in terms of security and performance.



References

Martin R. Albrecht, Benedikt Driessen, Elif Bilge Kavun, Gregor Leander, Christof Paar, and Tolga Yalçin.
Block ciphers - focus on the linear layer (feat. PRIDE).
pages 57–76, 2014.

Adnan Baysal and Sühap Sahin.
Roadrunner: A Small and Fast Bitslice Block Cipher for Low Cost 8-bit processors.
In Tim Güneysu, Gregor Leander, and Amir Moradi, editors, *LightSec 2015*, volume 9065, pages 58–76, Bochum, Germany, September 10-11, 2015.

Sylvain Guilley, Laurent Sauvage, Jean-Luc Danger, and Nidhal Selmane.
Fault injection resilience.
In Luca Breveglieri, Marc Joye, Israel Koren, David Naccache, and Ingrid Verbauwhede, editors, *FDTC 2010*, pages 51–65, Santa Barbara, California, USA, August 21, 2010. IEEE Computer Society.

Joe Kilian and Phillip Rogaway.
How to Protect DES Against Exhaustive Key Search, pages 252–267.
Springer Berlin Heidelberg, Berlin, Heidelberg, 1996.

Benjamin Lac, Marc Beunardeau, Anne Canteaut, Jacques Jean Alain Fournier, and Renaud Sirdey.
A First DFA on PRIDE: from Theory to Practice.
In *Proc. 11th International Conference on Risks and Security of Internet and Systems*, Roscoff, France, September 2016. Springer.

Ravikumar Selvam, Dillibabu Shanmugam, and Suganya Annadurai.
Side Channel Attacks: Vulnerability Analysis of PRINCE and RECTANGLE using DPA.
Cryptology ePrint Archive, Report 2014/644, 2014.
http://eprint.iacr.org/2014/644.

Elena Trichina.
Combinational logic design for aes subbyte transformation on masked data.
Technical report, IACR report, 2003.

Thank you for your time and attention!