esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

ФГУП «НПП «Гамма»

"KUZNYECHIK"

Optimized Implementations on FPGA and Microcontrollers and their DPA Analysis Resistance

Cédric DELAUNAY, ESIEA, (C + V) ^ O Lab, France

Alexander Alexandrovich ISTOMIN, "NPP "GAMMA", Russia

Eric FILIOL, ESIEA, (C + V) ^ O Lab, Laval, France

Делоне Седрик, ESIEA, (C+V)^O Lab, Франция

Истомин Александр Александрович, ФГУП «НПП «ГАММА», Россия

Фийол Эрик, ESIEA, (C+V)^O Lab, Лаваль, Франция

XXI - RUSCRYPTO 2019

20.03.2019

COOPERATION / КООПЕРАЦИЯ

ESIEA
(C + V) ^ O Lab.
Laboratory of Operational Cryptology and Virology,
France

FSUE "SIE "GAMMA"
Federal State Unitary Enterprise Scientific and Industrial Enterprise "GAMMA",
Russian Federation

ESIEA
(C + V) ^ O Lab.
Лаборатория оперативной Криптологии и Вирусологии,
Франция

ФГУП «НПП «ГАММА»
Федеральное Государственное Унитарное Предприятие «Научно-производственное предприятие «ГАММА»,
Россия

esiea
ECOLE D'INGENIEURS
DU MONDE NUMERIQUE

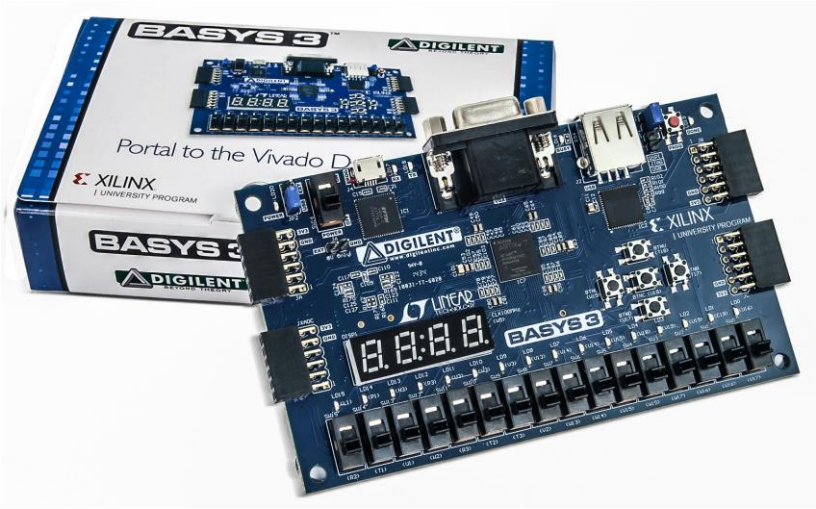
Laboratoire de cryptologie et de virologie opérationnelles - Corsaire pas pirate

Presentation Outline

1. **“Kuznyechik”** et al.
2. Securing the Implementation
3. **CPA** and **DPA** – Attacks and Resistance
4. Results and Conclusions
5. Where to next ?

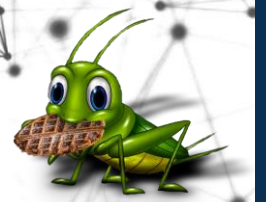
Initial context and stakes

- Study the symmetric **GOST R 34.12-2015** aka "**Grasshopper**" Block cipher algorithm.
- Provide different implementations for **FPGA** and **Microcontrollers**
- Check the resistance of the algorithm and its implementations to **CPA** and **DPA** attacks.



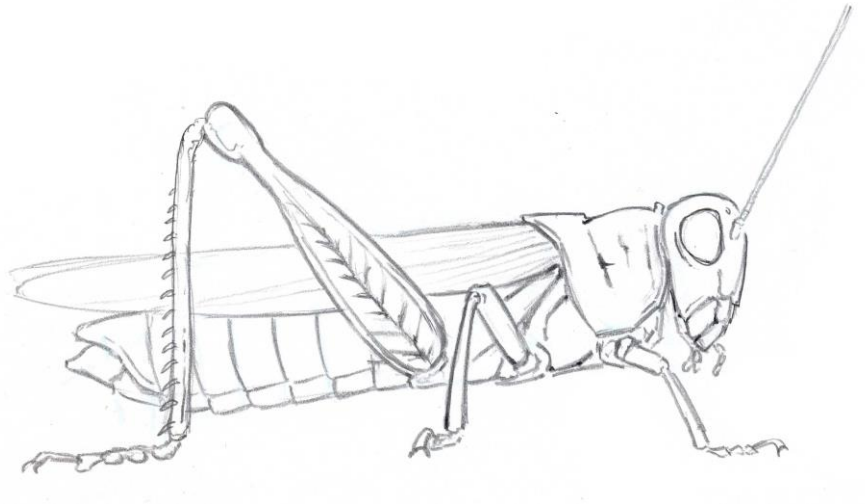
« Kuznyechik » et al.

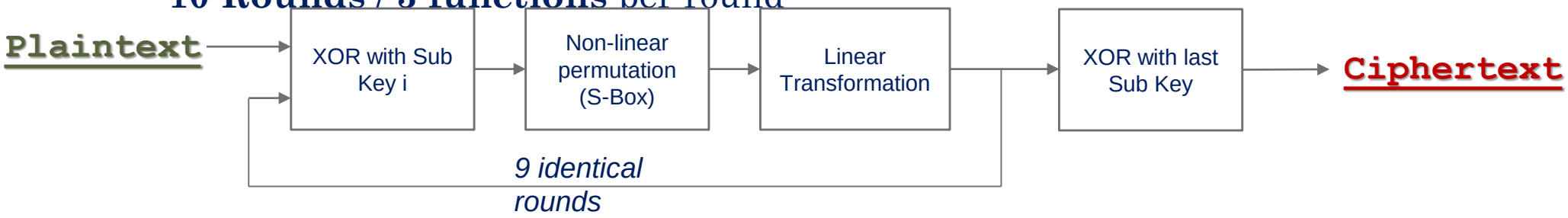
About Grasshoppers and Waffles



“Kuznyechik” (GOST R 34.12-2015)*

- **Russian Standard**, adopted 01/2016
- **Symmetric Block Cipher** algorithm
- **128 bits** Block size
- **256 bits** Key length
- **10 sub-keys** of **128 bits**
- Substitution-Permutation Network
- **Feistel Network** for Key Schedule
- **10 Rounds / 3 functions** per round





$$C = X[k_{10}]LSX[k_9]LSX[k_8]...LSX[k_2]LSX[k_1](P)$$

(*): for all those in hibernation for the last 5 years

“Kuznyechik” (GOST R 34.12-2015)*

- Linear permutation L :

where: $L(x) = R^{16}(x)$

$$R(x) = R(x_{15}||...||x_0) \qquad x = x_{15}||...||x_0$$
$$= l(x_{15}, ..., x_0)||x_{15}||x_1$$

and
where :

$$l(x_{15}, ..., x_0) = 148x_{15} + 32x_{14} + 133x_{13} + 16x_{12} + 194x_{11} + 192x_{10} + 1x_9 + 251x_8 + 1x_7 + 192x_6 + 194x_5 + 16x_4 + 133x_3 + 32x_2 + 148x_1 + 1x_0$$

where all multiplications are performed in $GF(2)[X]/p(x), with$
$p(x) = x^8 + x^7 + x^6 + x + 1 \in GF(2)[X]$

(*): for all those in hibernation for the last 5 years

The AES Cipher as brief reminder

- **NIST Standard**, adopted 2001
- **Symmetric Block Cipher** algorithm
- **128 bits** Block size
- **128, 192 or 256 bits** Key length
- **14 sub-keys** of **128 bits**
- Substitution-Permutation Network
- **14 Rounds / 4 functions** per round (for 256 bit key length)

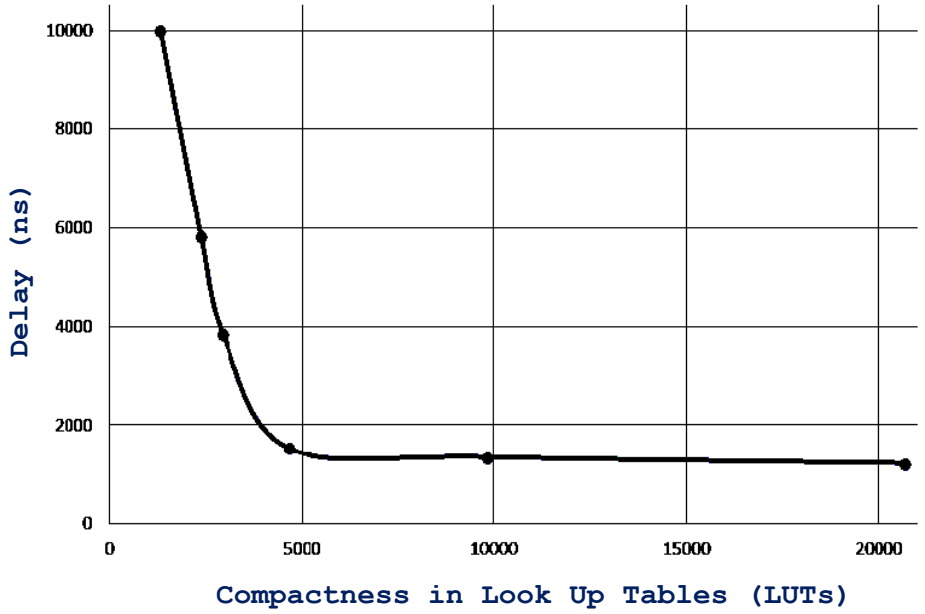
*

(*): Belgian Waffle – our avatar for the AES

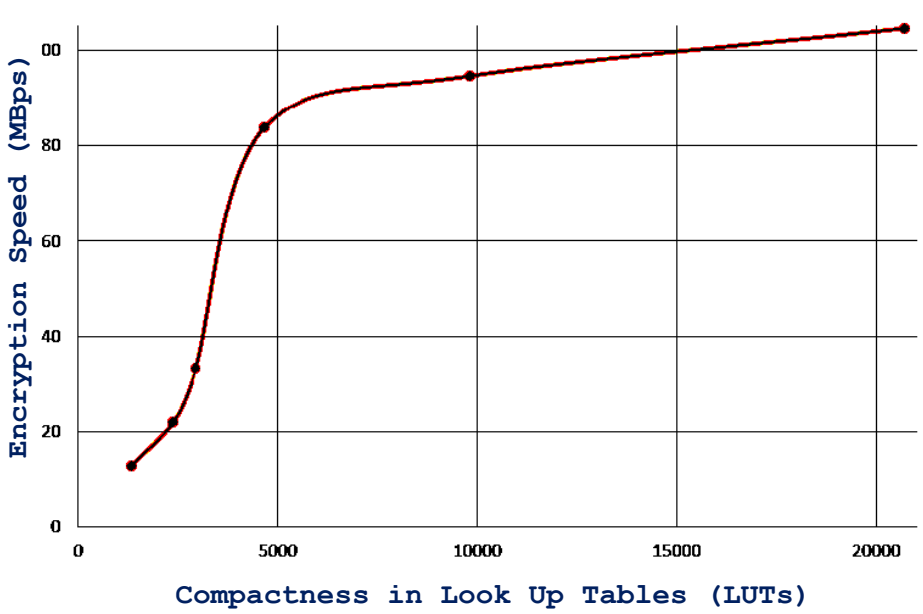
Kuznyechik on FPGA

- Several variants where conceptualized and implemented
- Here the results of the first implementations:

Encryption **DELAY** depending on the compactness of the model



Encryption **SPEED** depending on the compactness of the model



Securing the Implementation

Вежливые Зелёные Кузнечики

« It's not easy being green »

Threat Matrix

- **What do we face ?**

SIGINT / MEASINT

→ **Power Consumption Analysis - Attack**

[…] !!!

Fault Injection

- Voltage Glitch
- Temperature
- $\vec{h\nu}$ (Laser,…)

Microprobing

Side-Channel Analysis

Active

Passive

Invasive

Non - Invasive

HUMINT

Physical Attacks

Securing the Implementation

How can we protect ourselves against these attacks?

Enacting
HARDWARE and SOFTWARE
Countermeasures

- Random Delay Insertion
- Noise Generator
- Shuffling
- Masking
- Dual-Rail with Precharge Logic

Chosen Solution !

Securing the Implementation

How does the masked "Kuznyechik"-Implementation work ?

- Random Mask Generation
- XOR between the Plaintext and the mask before encryption (or decryption)
- Unmasking at the end of the encryption (or decryption)

$$\begin{aligned} LS_{m1}X[k](x+m) &= LS_{m1}(X[k](x)+m) \\ &= L(S(X[k](x))+m)) \\ &= LSX[k](x)+L(m) \end{aligned}$$

$$X[k_{10}]LS_{m9}X[k_9]...LS_{m1}X[k_1](x+m) = X[k_{10}]LSX[k_9]...LSX[k_1](x)+L^9(m)$$

Securing the Implementation

- **Masking AES-256** works the same way

$$
\begin{aligned}
A_{rk} M_c S_r S'_b(x+m) &= A_{rk} M_c S_r (S_b(x) + m) \\
&= A_{rk} M_c (S_r S_b(x) + S_r(m)) \\
&= A_{rk} (M_c S_r S_b(x) + M_c S_r(m)) \\
&= A_{rk} M_c S_r S_b(x) + M_c S_r(m)
\end{aligned}
$$

$$
\begin{aligned}
A_{rk} &: AddRoundKey \\
M_c &: MixColumns \\
S_r &: ShiftRows \\
S_b &: SubBytes
\end{aligned}
$$

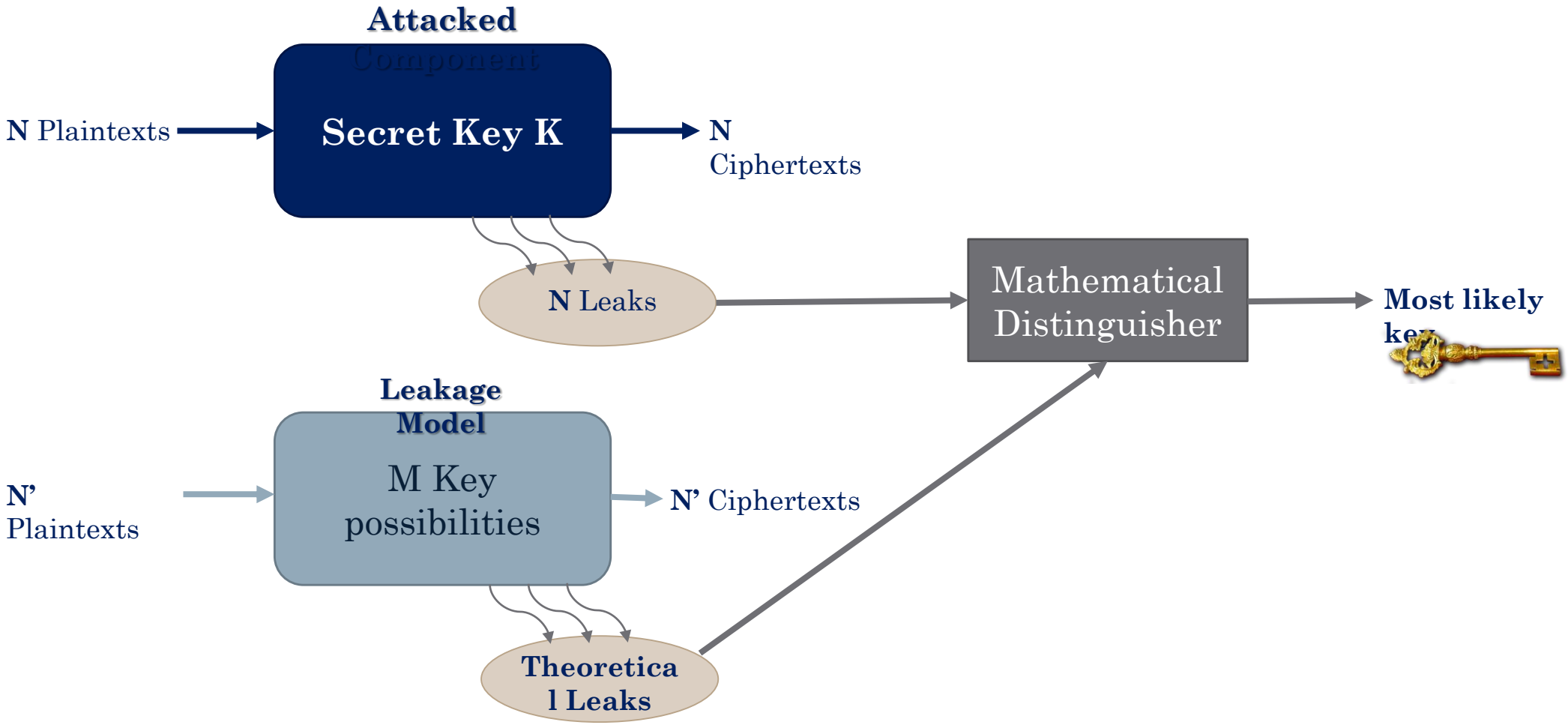
Securing the Implementation

- **Comparison between implementations:**
  (Reference Platform XILINX Artix7 (Speedgrade -1))

Implementation:	"Kuznyechik" Standard	"Kuznyechik" Masked	AES-256 Standard
LUTs:	8810	10106	4846
Latches:	2167	2697	4540
Maximum Frequency:	28.5 MHz	28.5 MHz	28.5 MHz
Delay:	**1526.2 ns**	**1596.2 ns**	1920.0 ns
Encryption rate	**83.9 Mbps**	**80.8 Mbps**	66.7 Mbps

Preview of the CPA Theory

- Leakage Model

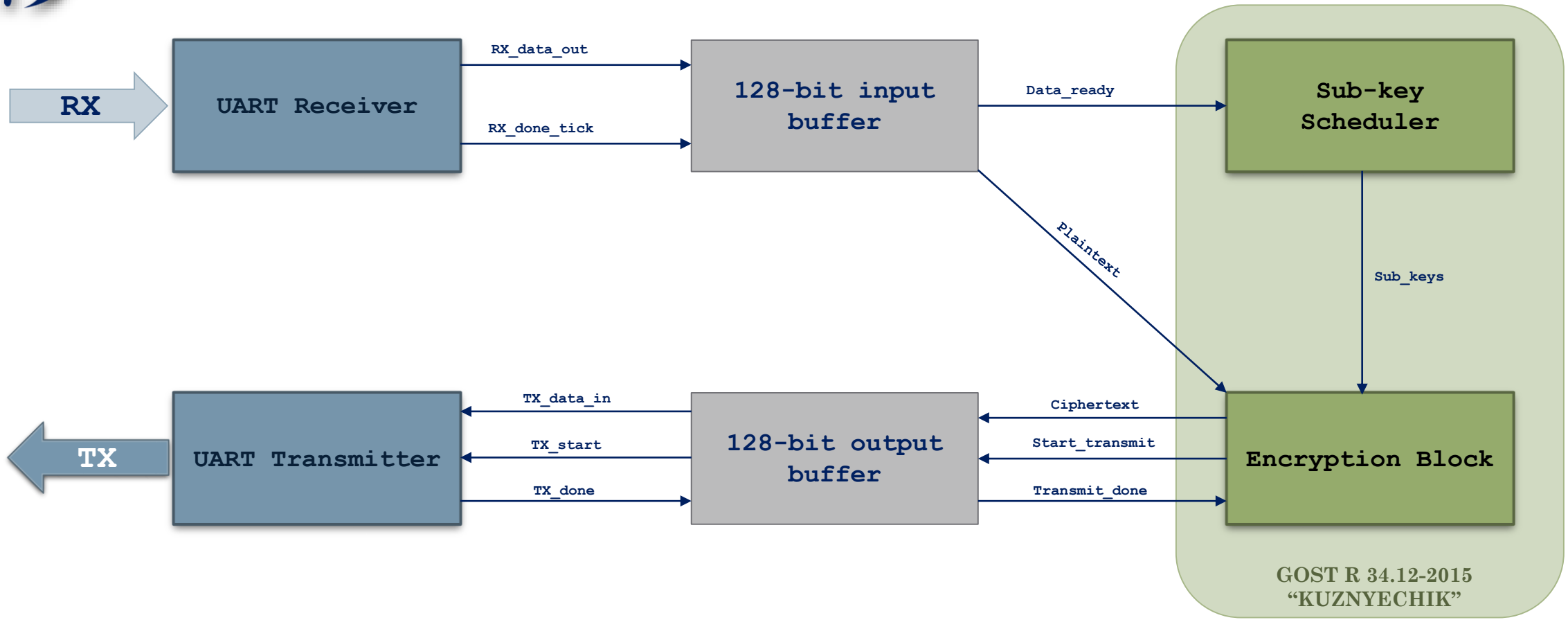


CPA and DPA Attacks and Resistance

…after that, the NSA, the CIA,…

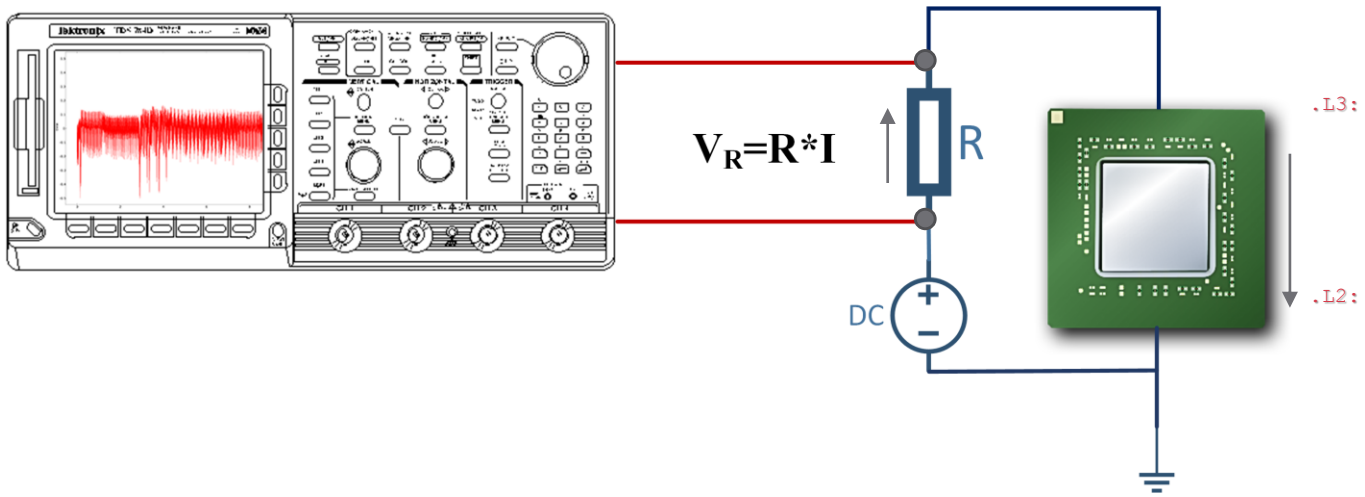
DPA Setup in FPGA

- XILINX Artix 7 XC7A35T & XC7A100T



Power Measurement

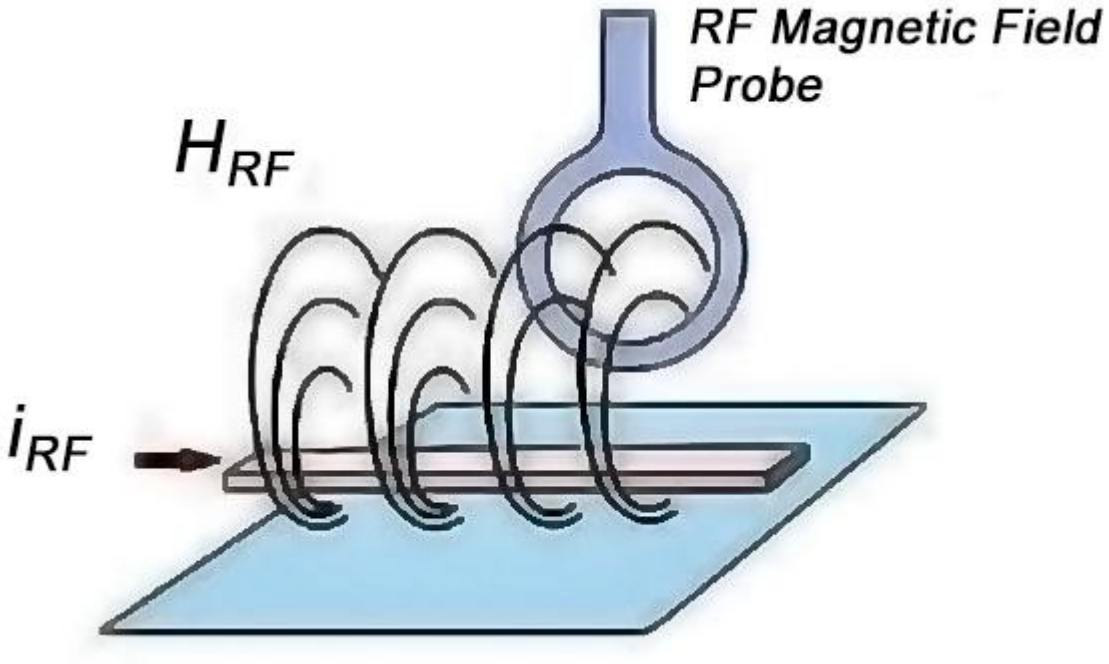
- Direct measurement at the source
- Usually through a shunt resistor in the Power supply line
  - The voltage at measured at the resistor is linear proportional to the current through the circuit



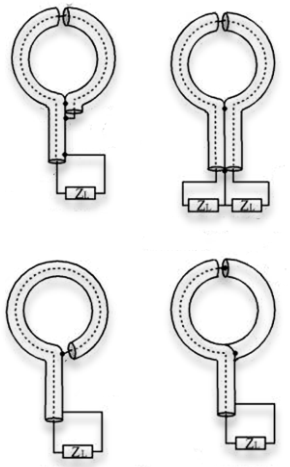

```
[…]
add fp, sp, #0
sub sp, sp, #20
str r0, [fp, #-16]
mov r3, #0
str r3, [fp, #-8]
b .L2

.L3:
ldr r3, [fp, #-8]
add r3, r3, #1
str r3, [fp, #-8]
ldr r3, [fp, #-16]
mov r3, r3, asr #1
str r3, [fp, #-16]

.L2:
ldr r3, [fp, #-16]
cmp r3, #0
bgt .L3
ldr r3, [fp, #-8]
mov r0, r3
add sp, fp, #0
ldmfd sp!, {fp}
bx lr
[…]
```

EM-Field Measurement

- Indirect measurement
- Current through conductor = magnetic field
- Measured through an **H-Probe**



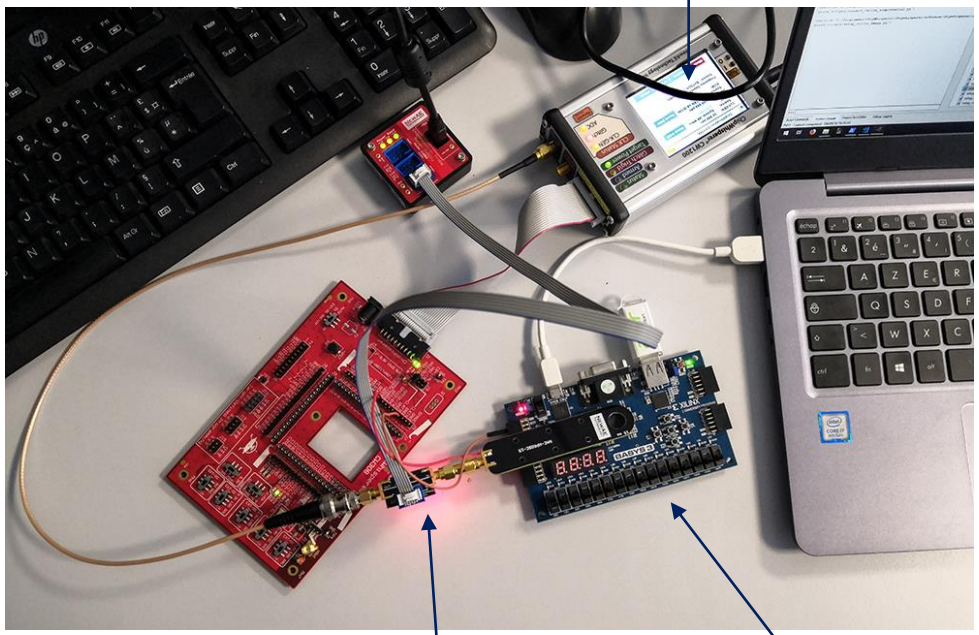
H-Probe Constructions :



Hardware Setup

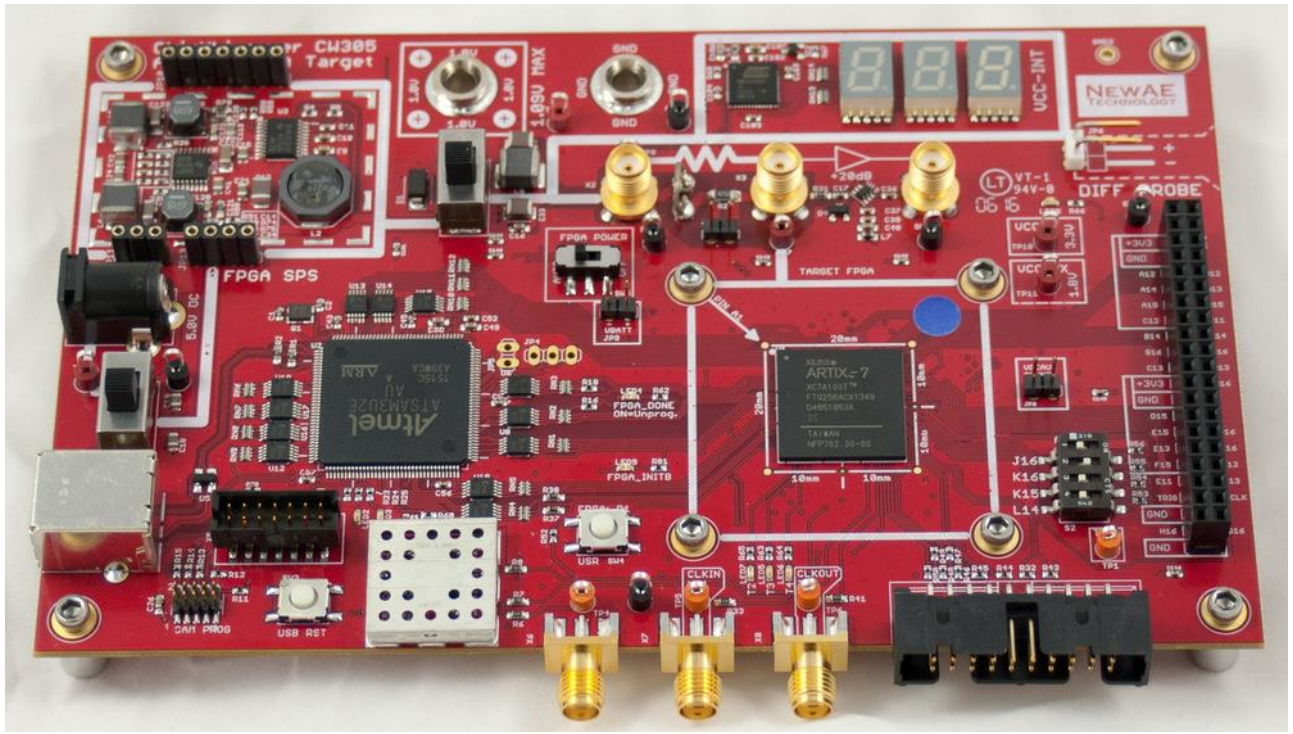
- Data Acquisition Equipment (France)
  - Power measurement through shunt on Microcontroller
  - Power measurement through H-Probe on Artix Devboard

ChipWhisperer CW1200



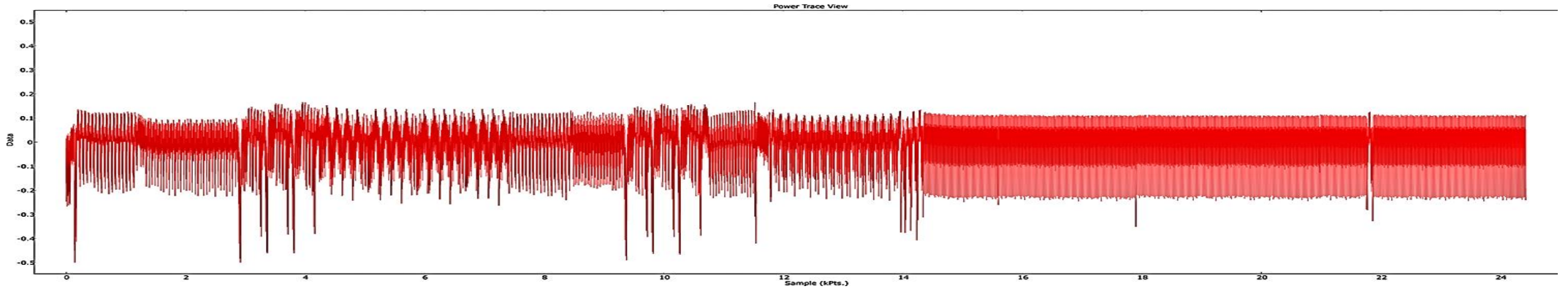
Hardware Setup

- Data Acquisition Equipment – Russia
- Two Labs with ChipWhisperer CW1173 and special FPGA Target Board
  - Power measurement through shunt on Microcontroller
  - Power measurement through shunt on low noise FPGA Board XC7A100T

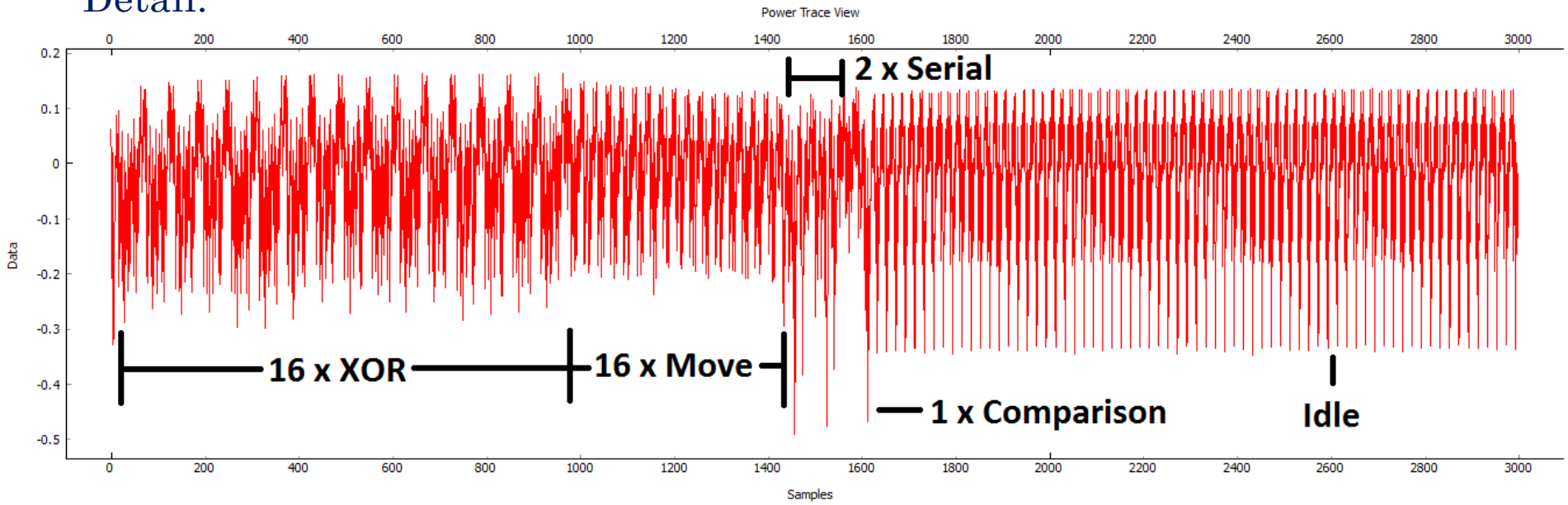


Standard CPA with AES (trivial)

Complete AES Signature (1 trace):

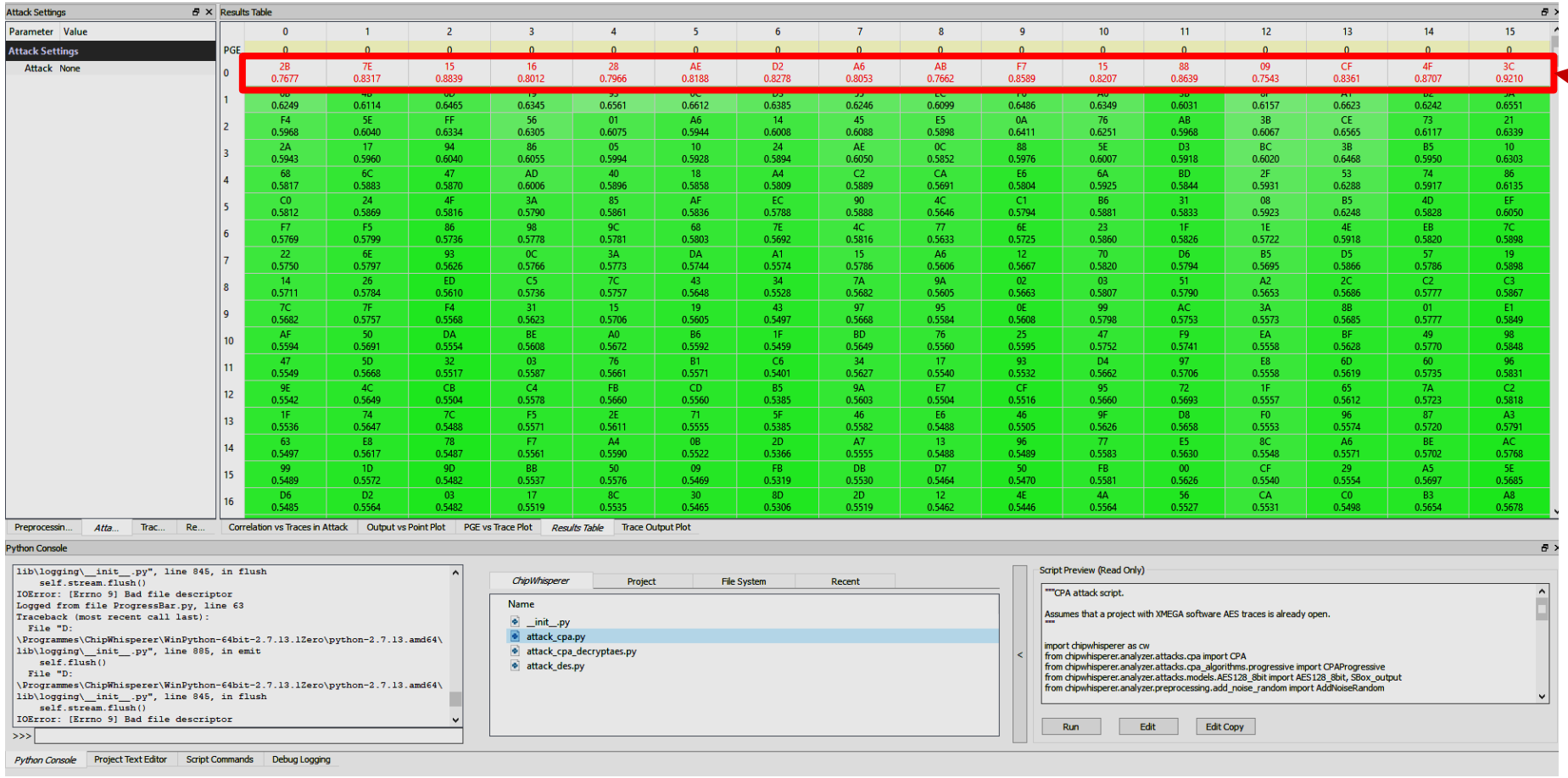


Detail:



Standard CPA with AES (trivial)

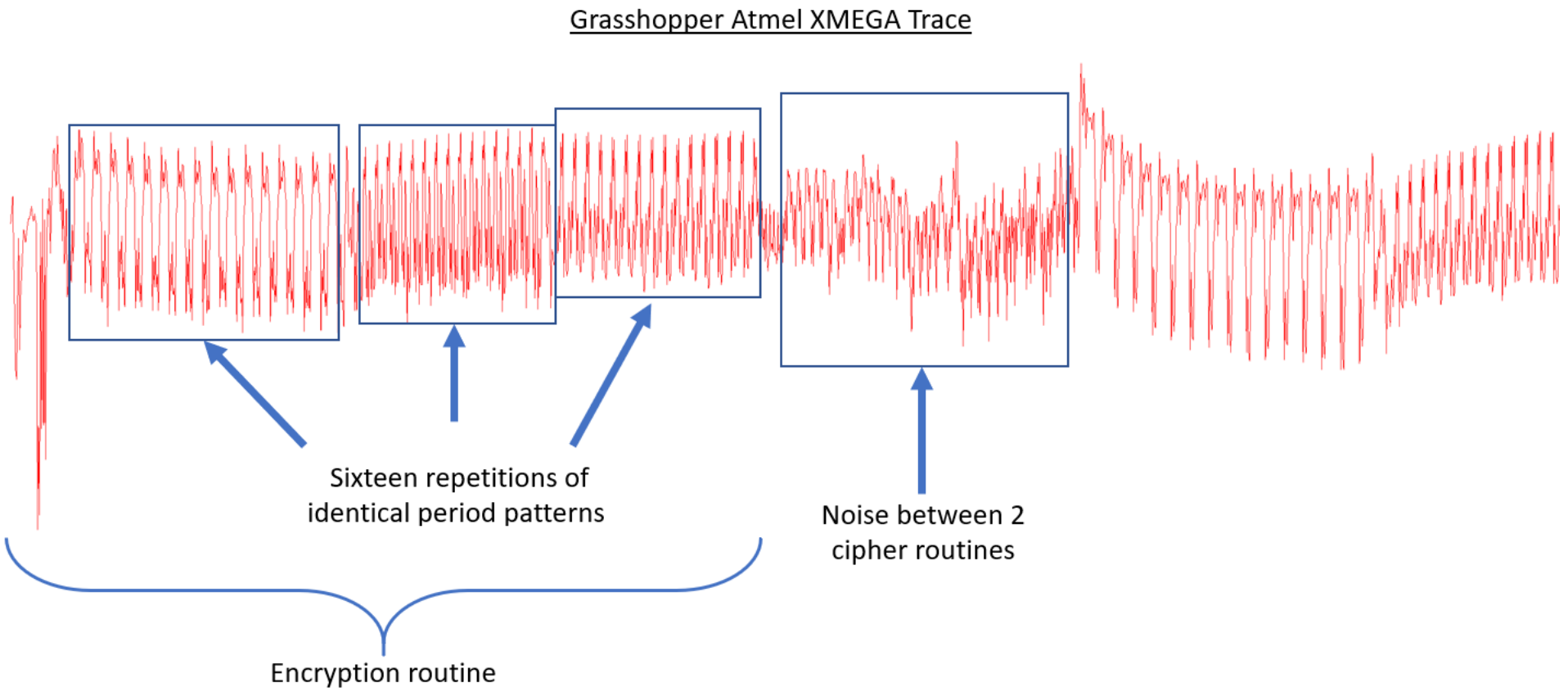
- An algorithm can be mathematically strong but very weak in hardware implementations without countermeasures



Key found with only 50 traces

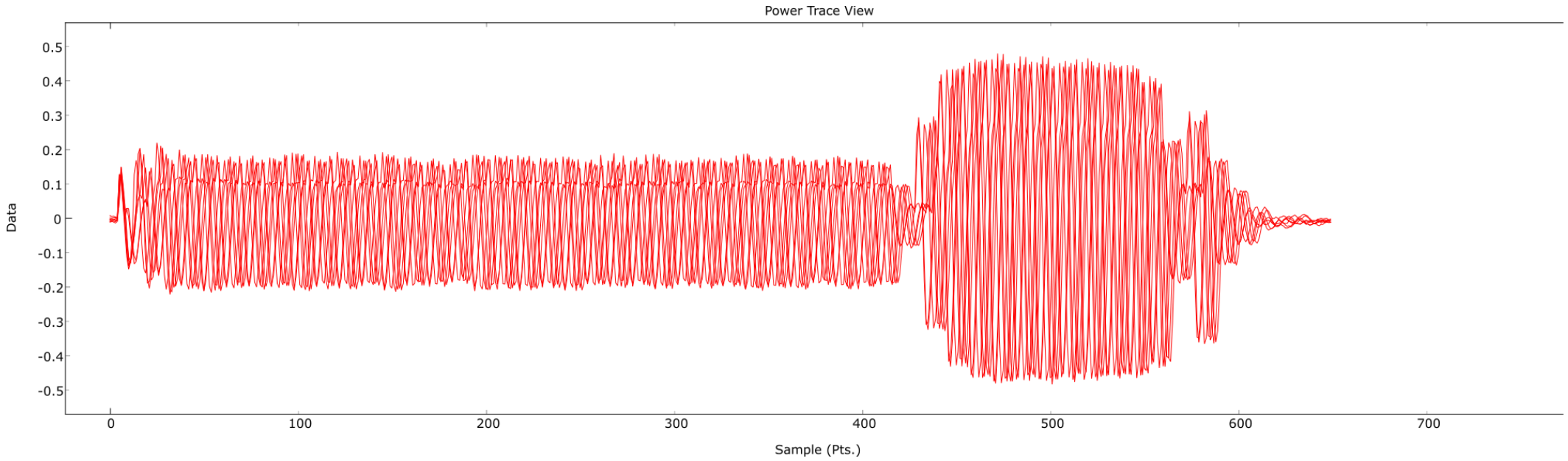
“Kuznyechik” Traces on Microcontroller

- Full trace



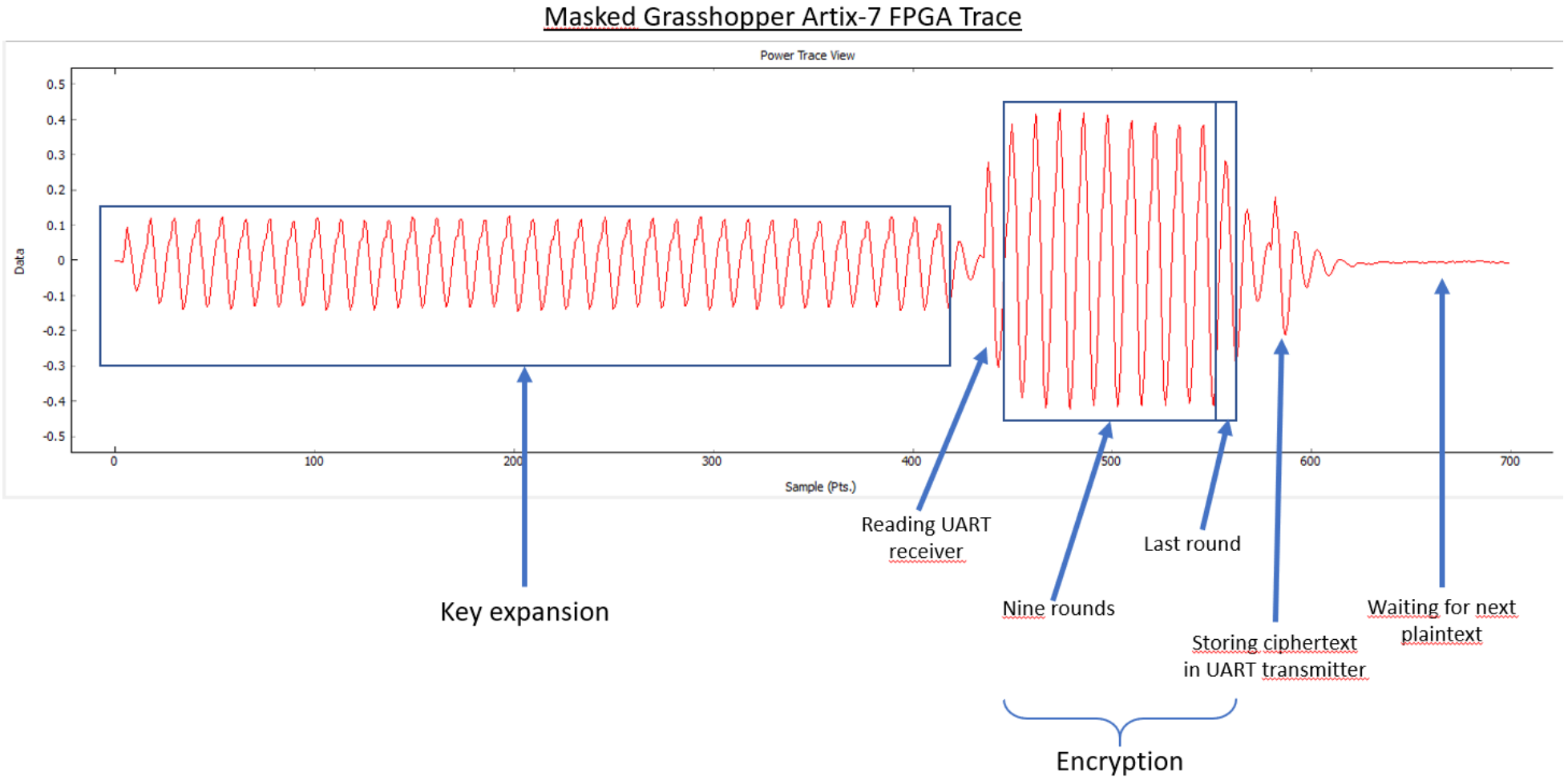
Masked – Kuznyechik on FPGA

- Masked / Non- Masked implementations have very similar signatures
- Here are 10+ traces superimposed (unsynchronized)



Masked – Kuznyechik on FPGA

- Annotated Single Trace



Results & Conclusions

Results

- Working set: 100 000 traces (4000 sample points / trace) acquired.
- Tested attacks:
  - **Rounds :** 1st round / 10th round / 9th round with 10th known key
  - **Leakage models :** Hamming Distance (DPA) & Hamming Weight (CPA)
  - **Mathematical Distinguisher :** Pearson correlation coefficient

Results

Despite the acquisition of a large number of traces, no subkeys could be extracted from a "**Kuznyechik**" encryption process. No "usual" model, working on **AES**, **DES**, **3DES** or **RSA** made it possible to determine one of the subkeys.

Even if a sub key were discovered, there would still be a problem:

- Subkeys are generated in pairs,
  so **<u>2 subkeys are needed to recover the master key</u>** (instead of one for AES !)

$$(K_{2i+1}, K_{2i+2}) = F[C_{8(i-1)+8}]...F[C_{8(i-1)+1}](K_{2i-1}, K_{2i}), \qquad i = 1, 2, 3, 4$$

$$F[k](a_1, a_0) = (LSX[k](a_1) \oplus a_0, a_1)$$

Where

Results

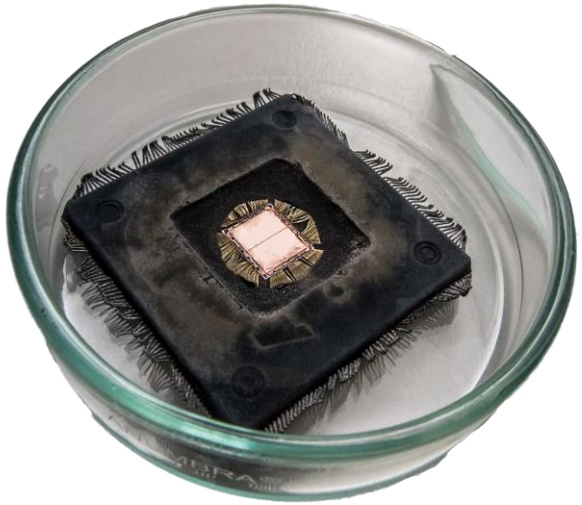
- Four optimized implementations (2x FPGA + 2x Microcontroller) have been developed, allowing to compare the “Kuznyechik” (GOST R 34.12-2015) algorithm and AES.
- However, even if security elements have been addressed, it is impossible to say whether or not this algorithm is sensitive to DPA / CPA attacks.
- Less “traditional” methods,
  such as attacks involving the use of **Machine Learning (ML)**, should be considered.
- Or maybe …

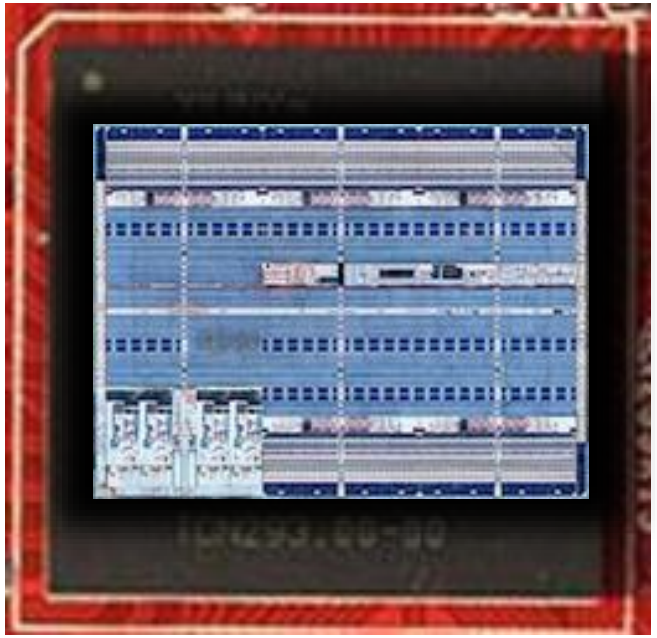
Where to next ?

~~Кто виноват ?~~ Что делать ?

3D - Leakage Cartography

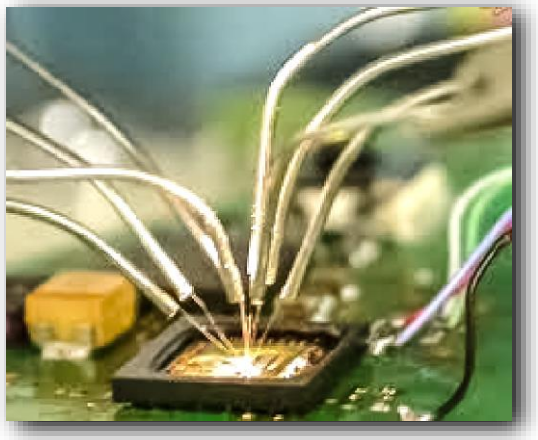
- 2D + Time !
- Works better with chip decapping !

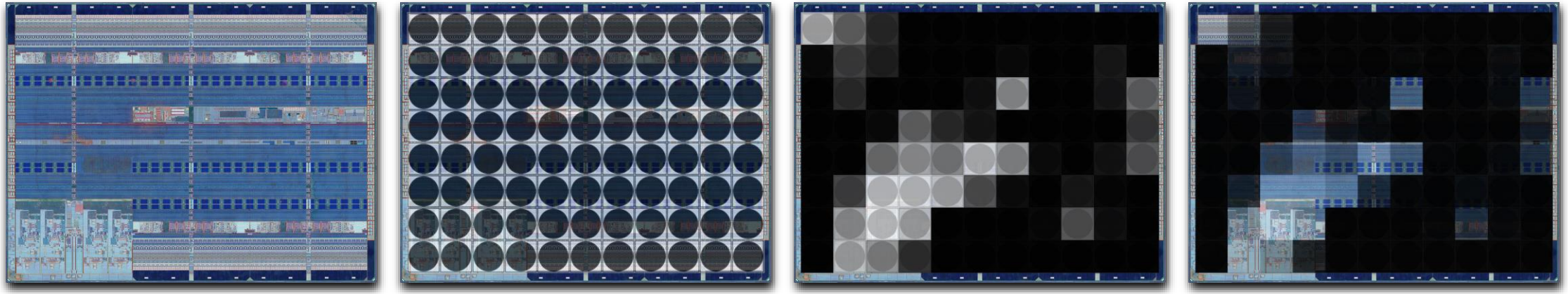




3D - Leakage Cartography

- Very precise X-Y Plotter Table (special Construction)
- Micro H-Probe / Micro H-Probe Array (ev. E-Probe)
- Work in progress !





Caveats ?

- Very High Analysis Cost ! Pessimistic : (X*Y : X*Y) * Traces
- Not very usable for small implementations
  - → Small attack surface
- Targeted for high speed implementations using most FPGA gate area
- Semi-invasive / full invasive technique (a little  !)

Thank you for your attention !

Cédric DELAUNAY,
ESIEA, (C + V) ^ O Lab, France
cedric.delaunay@ensta-bretagne.org

Alexander Alexandrovich ISTOMIN,
FSUE "SIE "GAMMA", Russia
istomin.a@nppgamma.ru

Eric FILIOL,
ESIEA, (C + V) ^ O Lab, Laval, France
e.filiol@esiea.fr

Делоне Седрик,
ESIEA, (C+V)^O Lab, Франция
cedric.delaunay@ensta-bretagne.org

Истомин Александр Александрович,
ФГУП «НПП «ГАММА», Россия
istomin.a@nppgamma.ru

Фийоль Эрик,
ESIEA, (C+V)^O Lab, Лаваль, Франция
e.filiol@esiea.fr

ФГУП «НПП «Гамма»