Side-Channel Analysis of Multiplications in GF(2^{128})

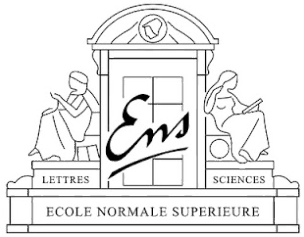
Application to AES-GCM

Sonia Belaïd[1] Pierre-Alain Fouque[2] Benoît Gérard[3]

[1]École normale supérieure and Thales Communications & Security,

[2]Université de Rennes 1 and Institut Universitaire de France

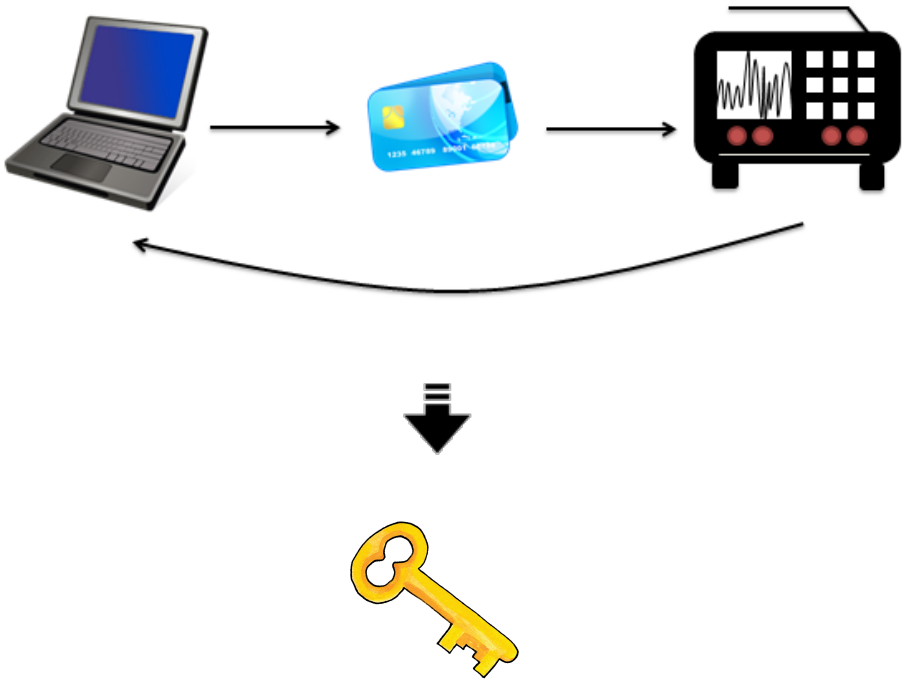
[3]DGA–MI and IRISA



THALES

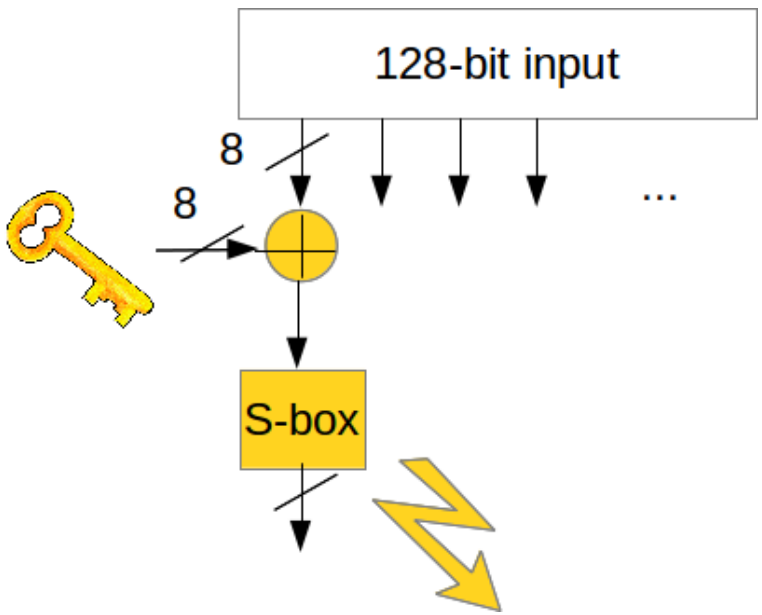
Side-Channel Attacks

- physical leakage
  - timing
  - power consumption
  - temperature
  - ...
- statistical treatment
- key recovery



Key-Dependent Leakage

AES Block Cipher



Attack on 8 bits

- prediction of the outputs for the 256 possible 8-bit secret
- correlation between predictions and leakage
- selection of the best correlation to find the correct 8-bit secret

Attack on 128 bits

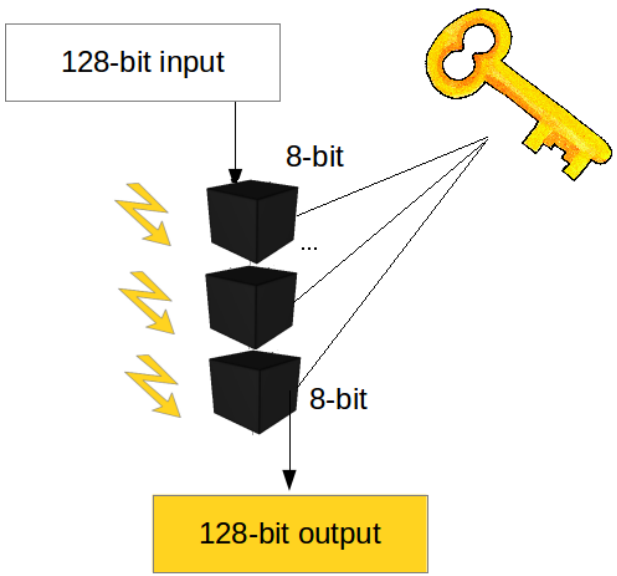
- repetition of the attack on 8 bits on each S-box

Key-Dependent Leakage

Multiplication in GF(2^{128}) (e.g., in the AES-GCM's authentication)

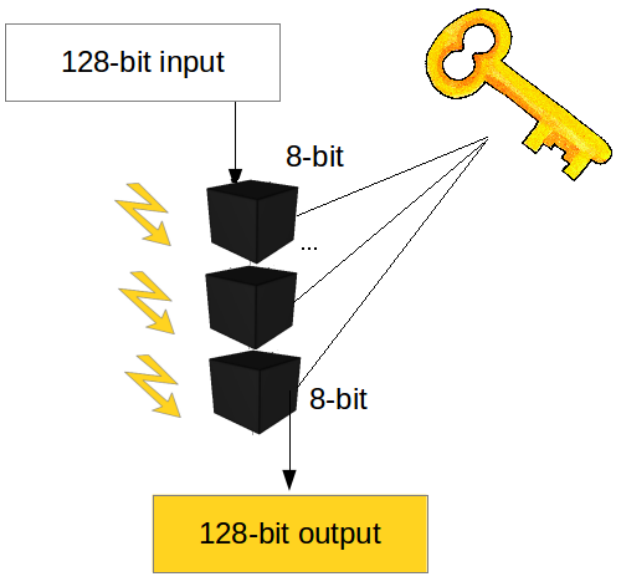
Key-Dependent Leakage

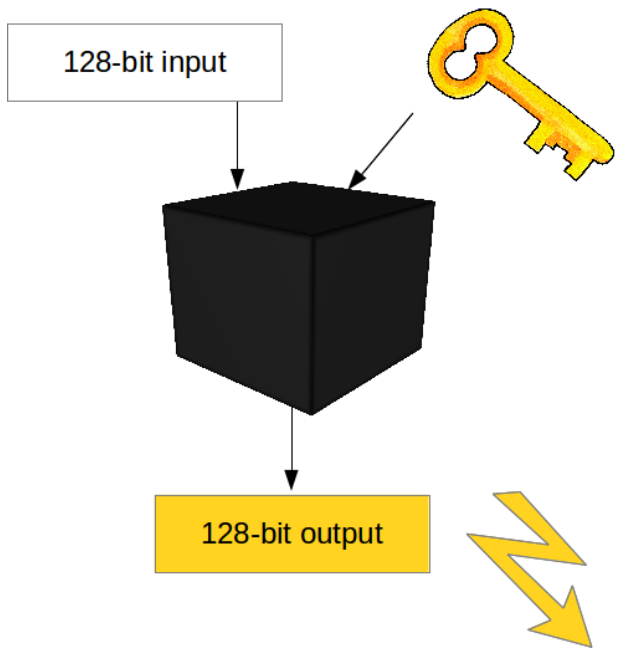
Multiplication in GF(2^{128}) (e.g., in the AES-GCM's authentication)



Key-Dependent Leakage

Multiplication in GF(2^{128}) (e.g., in the AES-GCM's authentication)



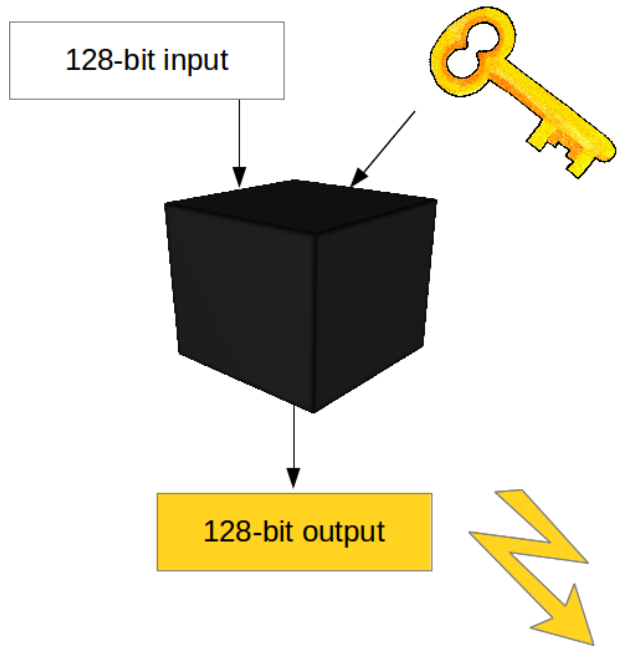


Contributions

Side-Channel Analysis of Multiplications in GF(2^{128}) : Application to AES-GCM

Sonia Belaïd, Pierre-Alain Fouque, Benoît Gérard

Asiacrypt 2014

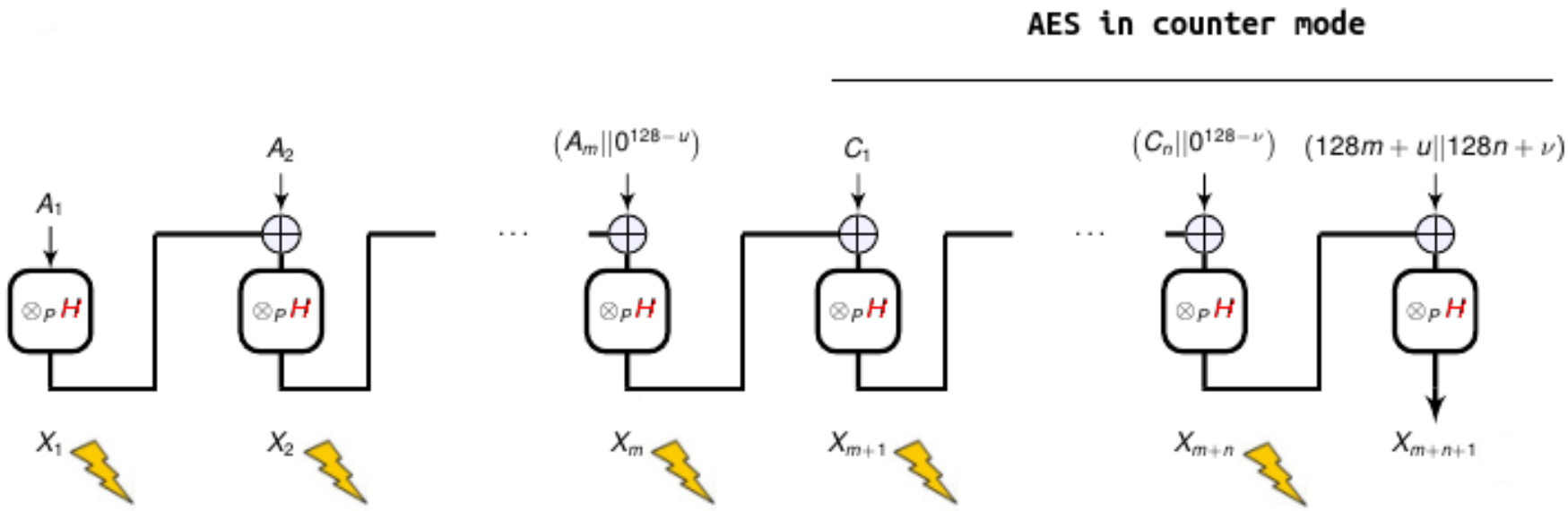


Outline

Outline

Outline

AES-GCM



secret: hashed key $H = \text{AES}_K(0^{128})$ with K the encryption key

inputs:
- 128-bit blocks of data to authenticate A_i
- 128-bit encrypted blocks C_i
  notation: M for either A_i or C_i

Galois Field Multiplication $\otimes_P$

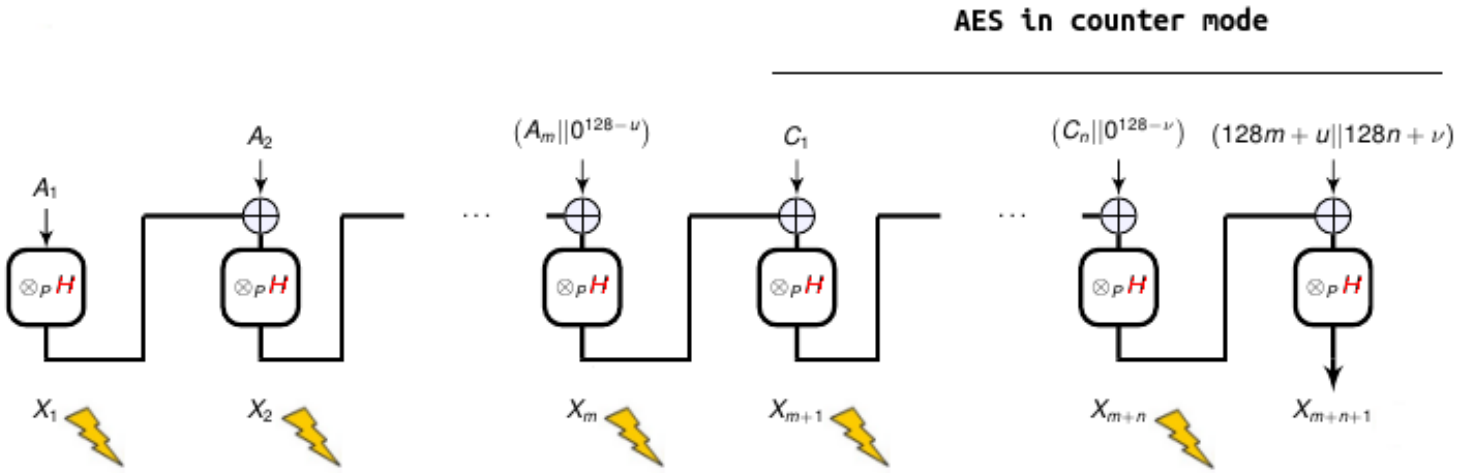
$$\mathrm{GF}(2^{128}) = \mathrm{GF}(2)[Y]/P(Y), \quad P(Y) = Y^{128} + Y^7 + Y^2 + Y + 1$$

$$M_P \cdot H =$$

$$\underbrace{\begin{pmatrix} m_0 \\ m_1 \\ \vdots \\ m_{127} \end{pmatrix}}_{M \otimes_P \alpha^0} \underbrace{\begin{pmatrix} m_{127} \\ m_0 \oplus m_{127} \\ \vdots \\ m_{126} \end{pmatrix}}_{M \otimes_P \alpha^1} \cdots \underbrace{\begin{pmatrix} m_1 \oplus m_{127} \oplus m_{126} \\ m_2 \oplus m_{123} \oplus m_1 \oplus m_{127} \oplus m_{122} \\ \vdots \\ m_0 \oplus m_{127} \oplus m_{126} \oplus m_{121} \end{pmatrix}}_{M \otimes_P \alpha^{127}} \begin{pmatrix} h_0 \\ h_1 \\ \vdots \\ h_{127} \end{pmatrix}$$

Outline

Leakage Models



Hamming Weight

$$L_i^{(\mathrm{HW})} = \mathrm{HW}(X_i) + \varepsilon_\sigma, \quad \varepsilon_\sigma \sim \mathcal{N}(0, \sigma)$$

Hamming Distance

$$L_i^{(\mathrm{HD})} = \mathrm{HD}(X_i, X_{i-1}) + \varepsilon_\sigma = \mathrm{HW}(X_i \oplus X_{i-1}) + \varepsilon_\sigma$$

Attacker Capabilities

Known/Chosen Inputs:

- ciphertexts
- authenticated data

Limited/Unlimited Queries:

- error-counter for the tag verifications

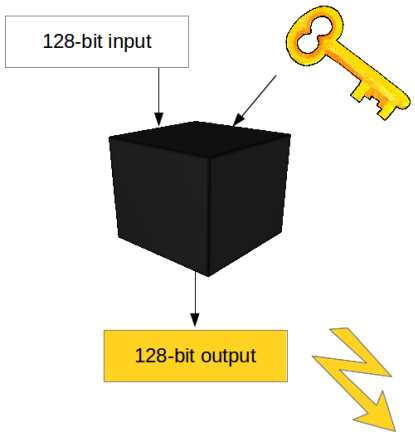
Outline

Outline

Main Idea of The Attack

Current Issue:
each bit of the 128-bit multiplication's result depends on all the key bits

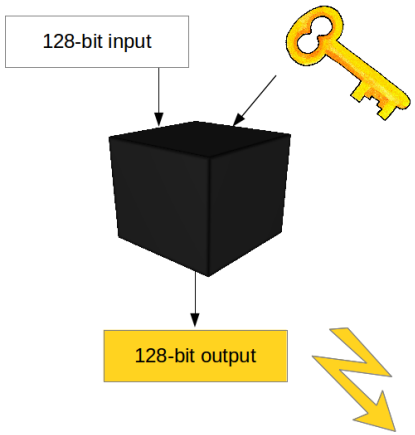
➜ no divide-and-conquer strategy



Main Idea of The Attack

Current Issue:
each bit of the 128-bit multiplication's result depends on all the key bits

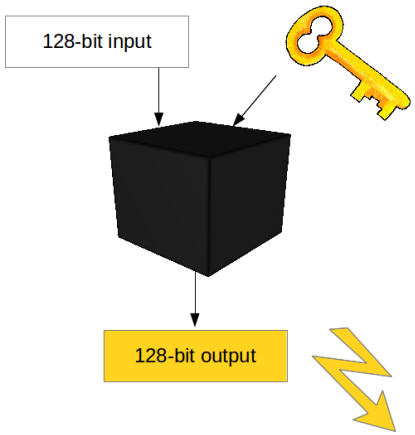
➜ no divide-and-conquer strategy



What we have:
the leakage of the first multiplication's output

$$\mathrm{HW}(M \otimes H) + \varepsilon_\sigma$$

Main Idea of The Attack



Current Issue:
each bit of the 128-bit multiplication's result depends on all the key bits

➜ no divide-and-conquer strategy

What we have:
the leakage of the first multiplication's output

$$\mathrm{HW}(M \otimes H) + \varepsilon_\sigma$$

Main observation:
the LSB of the Hamming Weight (same for HD) of a variable is a linear function of its bits:

$$\mathrm{lsb}_0\left(\mathrm{HW}(V)\right) = \bigoplus_{0 \leqslant i \leqslant 127} v_i$$

LSB of the first multiplication output's Hamming weight:

$$b_0 \stackrel{\text{def}}{=} \mathrm{lsb}_0\left(\mathrm{HW}(M \otimes_P H)\right) = \bigoplus_{0 \leqslant i \leqslant 127} (M \otimes_P H)_i$$

$$= \bigoplus_{0 \leqslant i \leqslant 127} \left(\bigoplus_{0 \leqslant j \leqslant 127-i} m_j \right) h_i$$

$$=$$

$$\begin{array}{c} \\ \oplus \\ \vdots \\ \oplus \end{array} \begin{pmatrix} m_0 h_0 & \oplus & m_{127} h_1 & \oplus & \cdots & (m_1 \oplus m_{127} \oplus m_{126}) h_{127} \\ m_1 h_0 & \oplus & (m_0 \oplus m_{127}) h_1 & \oplus & \cdots & (m_2 \oplus m_{123} \oplus m_1 \oplus m_{127} \oplus m_{122}) h_{127} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ m_{127} h_0 & \oplus & m_{126} h_1 & \oplus & \cdots & (m_0 \oplus m_{127} \oplus m_{126} \oplus m_{121}) h_{127} \end{pmatrix}$$

Linear system to solve:

$$\mathcal{S} = \begin{cases} \bigoplus_{0 \leqslant i \leqslant 127} \left(\bigoplus_{0 \leqslant j \leqslant 127-i} m_j^{(0)} \right) & h_i = {b_0}^{(0)} \\ \bigoplus_{0 \leqslant i \leqslant 127} \left(\bigoplus_{0 \leqslant j \leqslant 127-i} m_j^{(1)} \right) & h_i = {b_0}^{(1)} \\ \cdots & \\ \bigoplus_{0 \leqslant i \leqslant 127} \left(\bigoplus_{0 \leqslant j \leqslant 127-i} m_j^{(t-1)} \right) & h_i = {b_0}^{(t-1)} \end{cases}$$

Solution: hashed key H

New Issue

New Issue: leakage comes with noise

$$\begin{aligned} \widetilde{b_0} &\stackrel{\text{def}}{=} \mathrm{lsb}_0\left(\lceil \mathrm{HW}(M \otimes_P H) + \varepsilon_\sigma \rfloor\right) \\ &= \mathrm{lsb}_0\left(\mathrm{HW}(M \otimes_P H)\right) \oplus b_{\mathcal{N}} \end{aligned}$$

Probability of error on $b_{\mathcal{N}}$: $p_\sigma = 1 - \sum_{i=-\infty}^{\infty} \int_{2i-0.5}^{2i+0.5} \phi_\sigma(t)dt$

$$\begin{aligned} \sigma = 0.5 &\rightarrow p_\sigma = 0.31 \\ \sigma = 1 &\rightarrow p_\sigma = 1/2 - 4.6\ 10^{-3} \\ \sigma = 2 &\rightarrow p_\sigma = 1/2 - 1.7\ 10^{-9} \\ \sigma \geqslant 3 &\rightarrow p_\sigma = 1/2 - \varepsilon \end{aligned}$$

Application on the other bits ?

$$b_i = \bigoplus_{0 \leqslant j_1 < \cdots < j_{2^i} \leqslant 127} \left(\prod_{1 \leqslant \ell \leqslant 2^i} \bigoplus_{0 \leqslant k \leqslant 127} (M \otimes_P \alpha^k)_{j_\ell} \; h_k \right), \quad \forall\, 0 \leqslant i \leqslant 7$$

σ	Bernoulli parameter p							
	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7
0.5	$3.1\,10^{-1}$	$1.6\,10^{-1}$	$8.0\,10^{-2}$	$4.0\,10^{-2}$	$2.3\,10^{-2}$	$2.2\,10^{-2}$	$2.2\,10^{-2}$	ε
1	$\frac{1}{2} - 4.6\,10^{-3}$	$3.7\,10^{-1}$	$1.9\,10^{-1}$	$9.5\,10^{-2}$	$5.5\,10^{-2}$	$5.3\,10^{-2}$	$5.3\,10^{-2}$	ε
2	$\frac{1}{2} - 1.5\,10^{-4}$	$\frac{1}{2} - 3.2\,10^{-3}$	$3.8\,10^{-1}$	$2.0\,10^{-1}$	$1.1\,10^{-1}$	$1.1\,10^{-1}$	$1.1\,10^{-1}$	ε
3	$\frac{1}{2} - \varepsilon$	$\frac{1}{2} - 6.8\,10^{-8}$	$4.7\,10^{-1}$	$3.0\,10^{-1}$	$1.6\,10^{-1}$	$1.5\,10^{-1}$	$1.5\,10^{-1}$	ε
4	$\frac{1}{2} - \varepsilon$	$\frac{1}{2} - 1.2\,10^{-9}$	$\frac{1}{2} - 3.0\,10^{-3}$	$3.8\,10^{-1}$	$2.1\,10^{-1}$	$1.9\,10^{-1}$	$1.9\,10^{-1}$	ε
5	$\frac{1}{2} - \varepsilon$	$\frac{1}{2} - \varepsilon$	$\frac{1}{2} - 1.9\,10^{-4}$	$4.4\,10^{-1}$	$2.6\,10^{-1}$	$2.3\,10^{-1}$	$2.3\,10^{-1}$	ε

Outline

Naive Attack

$$\widetilde{\mathcal{S}} = \begin{cases} \bigoplus_{0 \leqslant i \leqslant 127} \left(\bigoplus_{0 \leqslant j \leqslant 127-i} m_j^{(0)} \right) & h_i = \widetilde{b_0}^{(0)} \\ \bigoplus_{0 \leqslant i \leqslant 127} \left(\bigoplus_{0 \leqslant j \leqslant 127-i} m_j^{(1)} \right) & h_i = \widetilde{b_0}^{(1)} \\ \cdots & \\ \bigoplus_{0 \leqslant i \leqslant 127} \left(\bigoplus_{0 \leqslant j \leqslant 127-i} m_j^{(t-1)} \right) & h_i = \widetilde{b_0}^{(t-1)} \end{cases}$$

Naive attack

i) extract 128 equations linearly independent

ii) remove the errors on bits $\widetilde{b_0}^{(\ell)}$ by enumeration

Improved Attack

Complexities to improve

- number of queries/samples $\mathcal{C}_s$
- solving time $\mathcal{C}_t$

Methods

1. Reducing the Noise Impact $\longrightarrow$ to decrease (mainly) $\mathcal{C}_t$
2. Saving Executions $\longrightarrow$ to decrease $\mathcal{C}_s$
3. Solving the System with Dedicated Algorithms $\longrightarrow$ to decrease $\mathcal{C}_s$ and $\mathcal{C}_t$

1. Reducing the Noise Impact

First Idea: use the LLR (Log Likelihood Ratio) to approximate better the bit value b_0

$$\widehat{b_0} \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } \mathrm{LLR}(\ell) \geqslant 0, \\ 1 & \text{otherwise.} \end{cases}$$

with

$$\mathrm{LLR}(\ell) = \log(\ \mathbb{P}[b_0 = 0 \mid \ell]\ ) - \log(\ \mathbb{P}[b_0 = 1 \mid \ell]\ )$$

Second Idea: when more than 128 traces are available, choose 128 linearly independent samples from the *highest LLR absolute values*

Example:

if $\ell_1 = 64.01$ and $\ell_2 = 64.49$

➜ $|LLR(\ell_1)| > |LLR(\ell_2)|$

➜ we choose ℓ_1 and we throw away ℓ_2

1. Reducing the Noise Impact

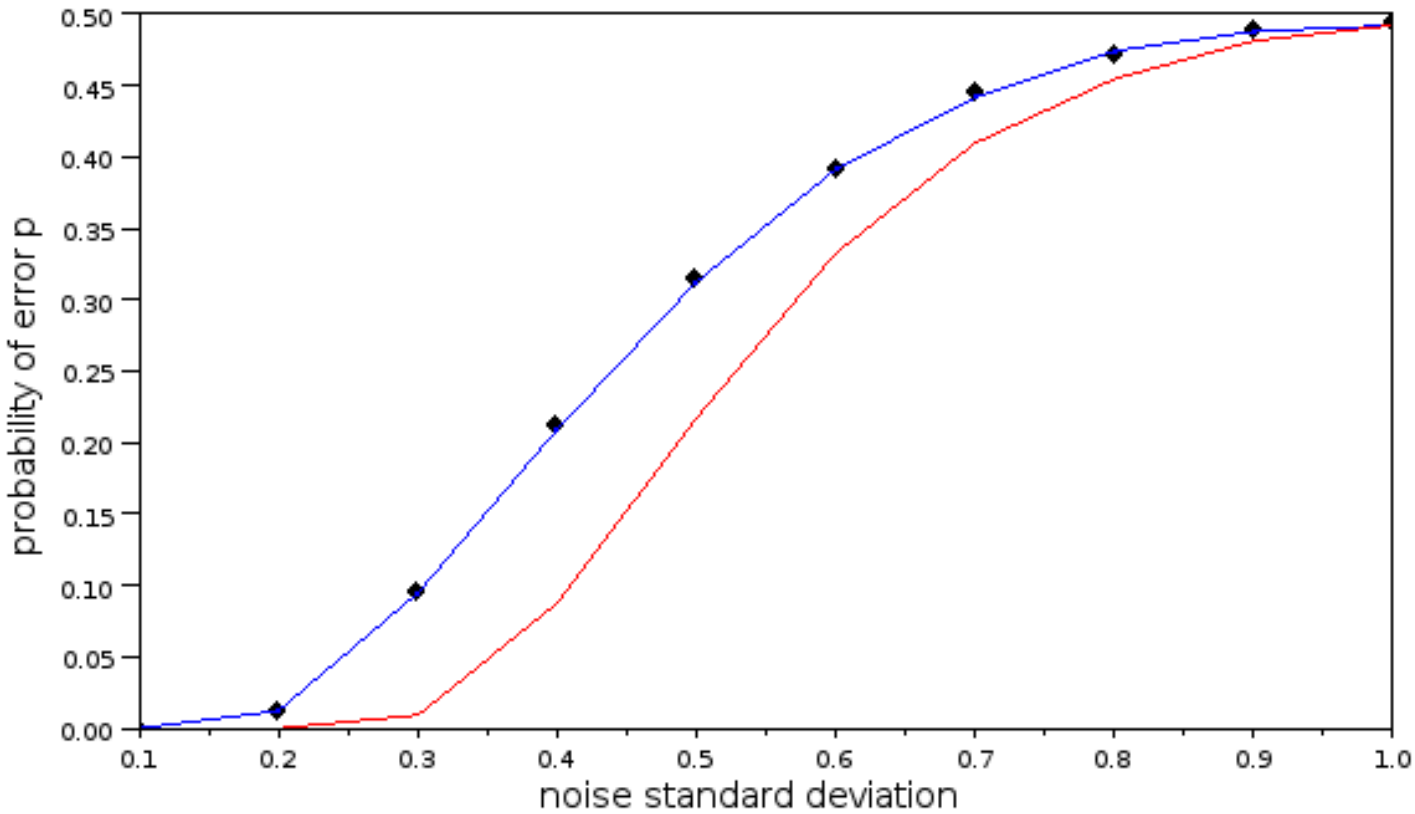
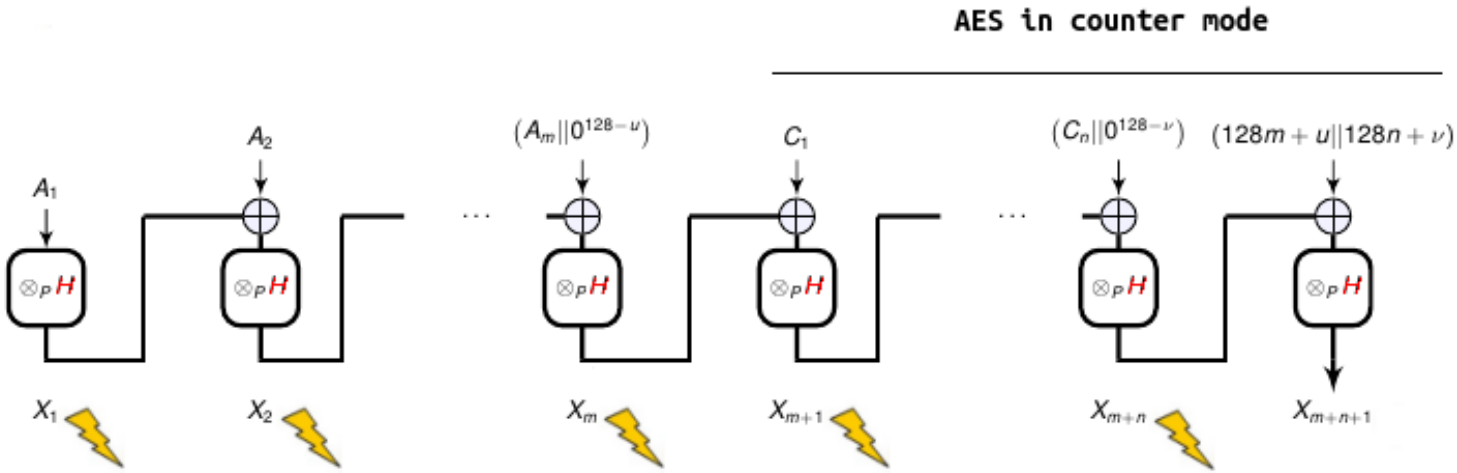


Figure: Error probability with rounding (black), LLR (blue) and best 128 LLRs (red) over 500 measurements

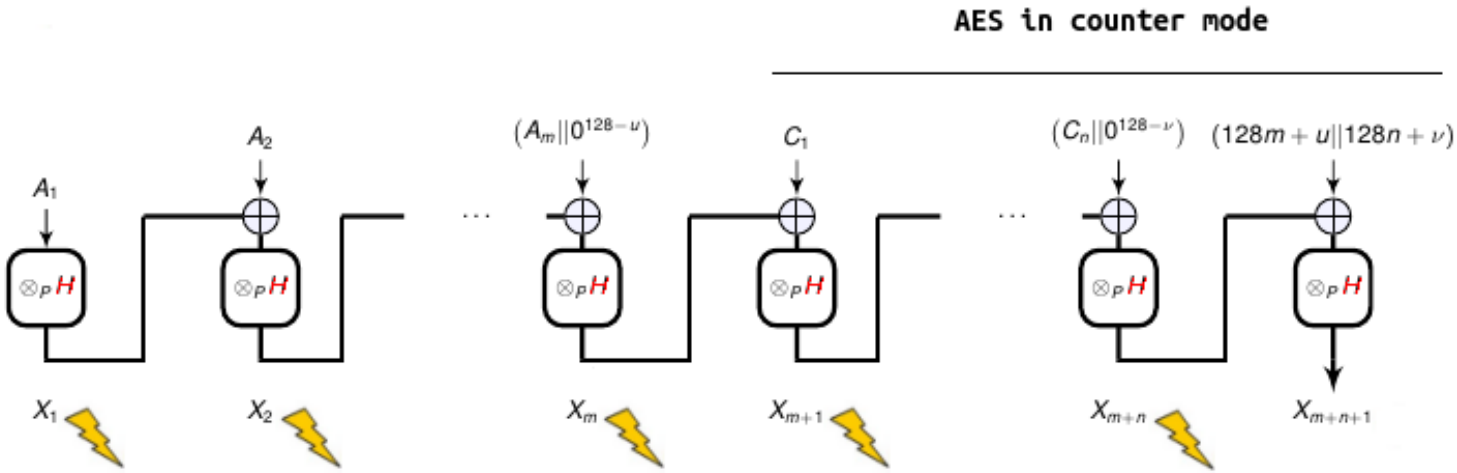
2. Saving Executions



Second Multiplication:

$$\begin{aligned} X_2 &= (M_1 \otimes_P H \oplus M_2) \otimes_P H \\ &= M_1 \otimes_P H^2 \oplus M_2 \otimes_P H \end{aligned}$$

2. Saving Executions



Second Multiplication:

$$\begin{aligned} X_2 &= (M_1 \otimes_P H \oplus M_2) \otimes_P H \\ &= M_1 \otimes_P H^2 \oplus M_2 \otimes_P H \end{aligned}$$

Since squaring is linear over GF(2), there exists S such that

$$X_2 = (M_1 \otimes_P S \oplus M_2) \otimes_P H$$

- two multiplications with a single execution : $\mathcal{C}_s \leftarrow \mathcal{C}_s/2$

3. Solving the System with Dedicated Algorithms

Noisy codeword: LSBs extracted from leaking multiplications that encode the authentication key H

Issue: decoding the noisy codeword

- Learning Parities with Noise (LPN) Algorithms
- Linear Decoding

Method \ σ	0.1 $\mathcal{C}_s/\mathcal{C}_t$	0.2 $\mathcal{C}_s/\mathcal{C}_t$	0.3 $\mathcal{C}_s/\mathcal{C}_t$	0.4 $\mathcal{C}_s/\mathcal{C}_t$	0.5 $\mathcal{C}_s/\mathcal{C}_t$
LLR + naive	$2^8/2^{21}$	$2^8/2^{21}$	$2^8/2^{22}$	$2^8/2^{65}$	$2^8/2^{107}$
LPN (LF Algo)	$2^{11}/2^{14}$	$2^{20}/2^{22}$	$2^{26}/2^{28}$	$2^{32}/2^{34}$	$2^{48}/2^{50}$
Linear decoding	$2^6/2^6$	$2^6/2^7$	$2^7/2^{11}$	$2^8/2^{25}$	$2^9/2^{62}$

Outline

Improvements

Complexities to improve

- number of queries/samples $\mathcal{C}_s$
- solving time $\mathcal{C}_t$

Methods

1. Averaging the traces $\longrightarrow$ to decrease (mainly) $\mathcal{C}_t$
2. Structuring the messages to make the system easier to solve $\longrightarrow$ to decrease (mainly) $\mathcal{C}_t$
3. Saving more executions $\longrightarrow$ to decrease $\mathcal{C}_s$

1. Averaging Traces

Repeating the same computation λ times: $\sigma \mapsto \sigma/\sqrt{\lambda}$

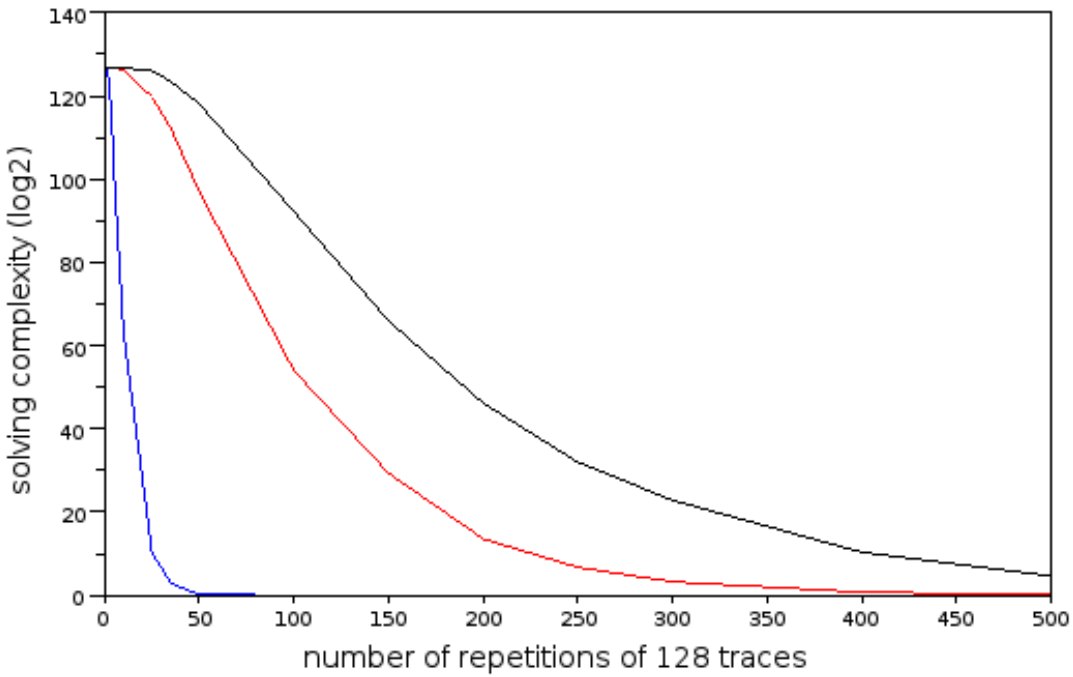


Figure: Solving complexities (naive attack) with repetitions for $\sigma = 1$ (blue), $\sigma = 3$ (red) and $\sigma = 4$ (black)

Experimental Results: tests on the Virtex-5 FPGA of a SASEBO board with an EM probe for the acquisition

➜ confirm the simulations

2. Structuring the Messages

Current Issue: the linear code corresponding to our attack is random and have a high dimension (128)

Better Code: concatenation of smaller random linear codes

- with the enumeration algorithm from [1], an attacker can enumerate keys from ordered lists of key chunks
- each block corresponds to a smaller linear code that may be fully decoded by a Fast Walsh Transform.

$$\begin{pmatrix} \boxed{\mathcal{S}_0} & & \\ & \boxed{\mathcal{S}_1} & \\ & & \ddots \end{pmatrix} \cdot \begin{pmatrix} H \end{pmatrix} = \begin{pmatrix} \widehat{b}_0 \\ \vdots \\ \widehat{b}_t \end{pmatrix}$$

[1] Veyrat-Charvillon, Gérard, Renauld, and Standaert. *An optimal key enumeration algorithm and its application to side-channel attacks.* In SAC 2012, LNCS, pp 390–406.

2. Structuring the Messages

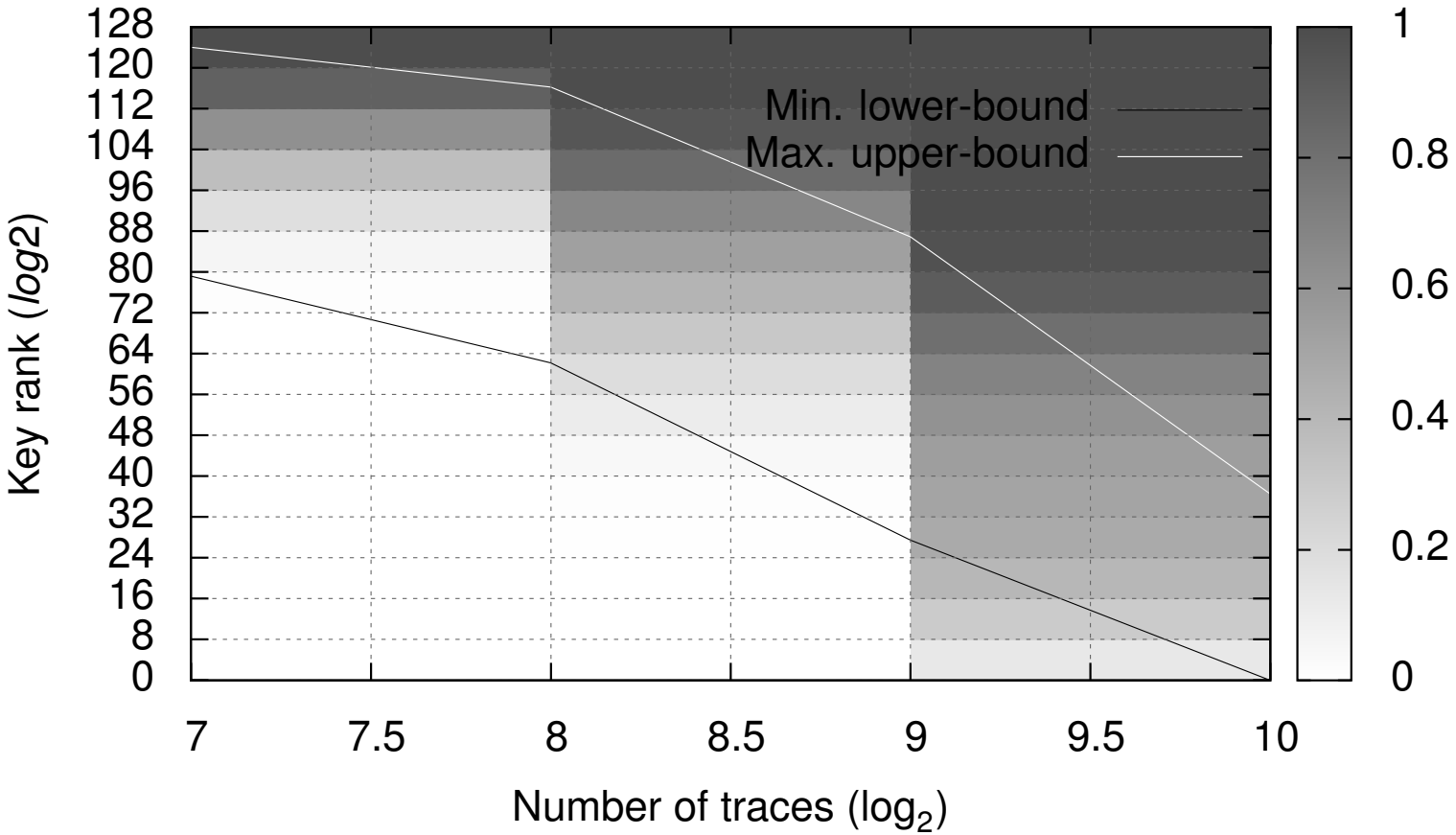
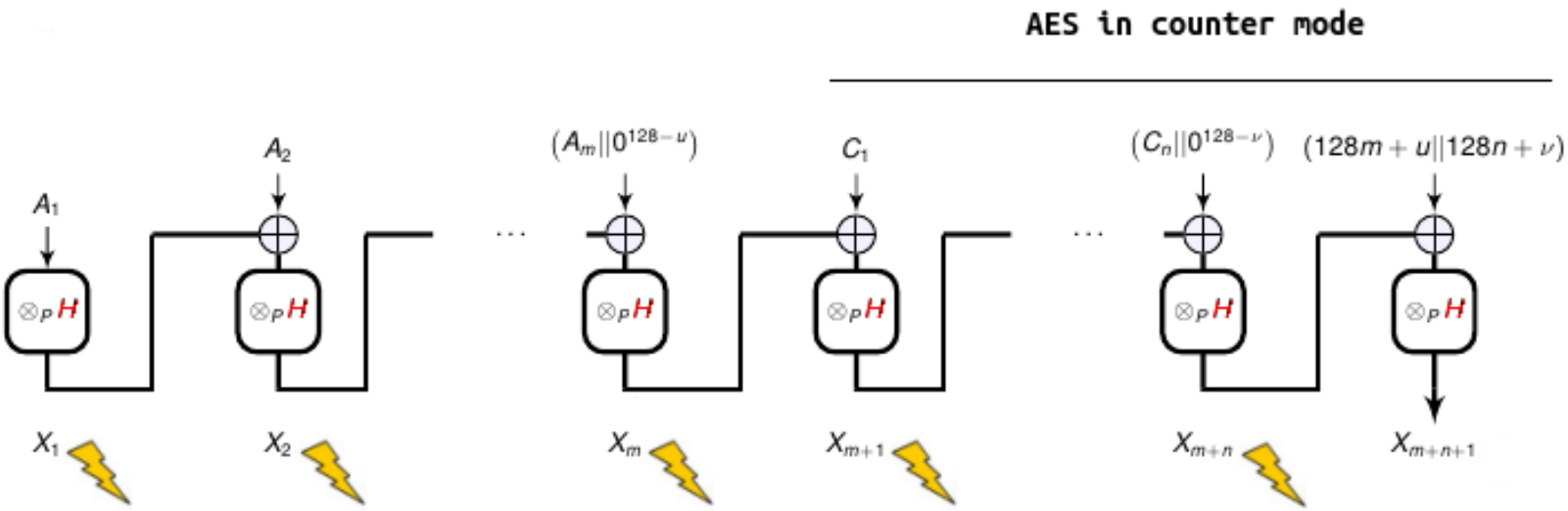


Figure: Security graph[2] for $\sigma = 0.5$

[2] Veyrat-Charvillon, Gérard, and Standaert. *Security evaluations beyond computing power.* In EUROCRYPT 2013, LNCS, pp 126–141.

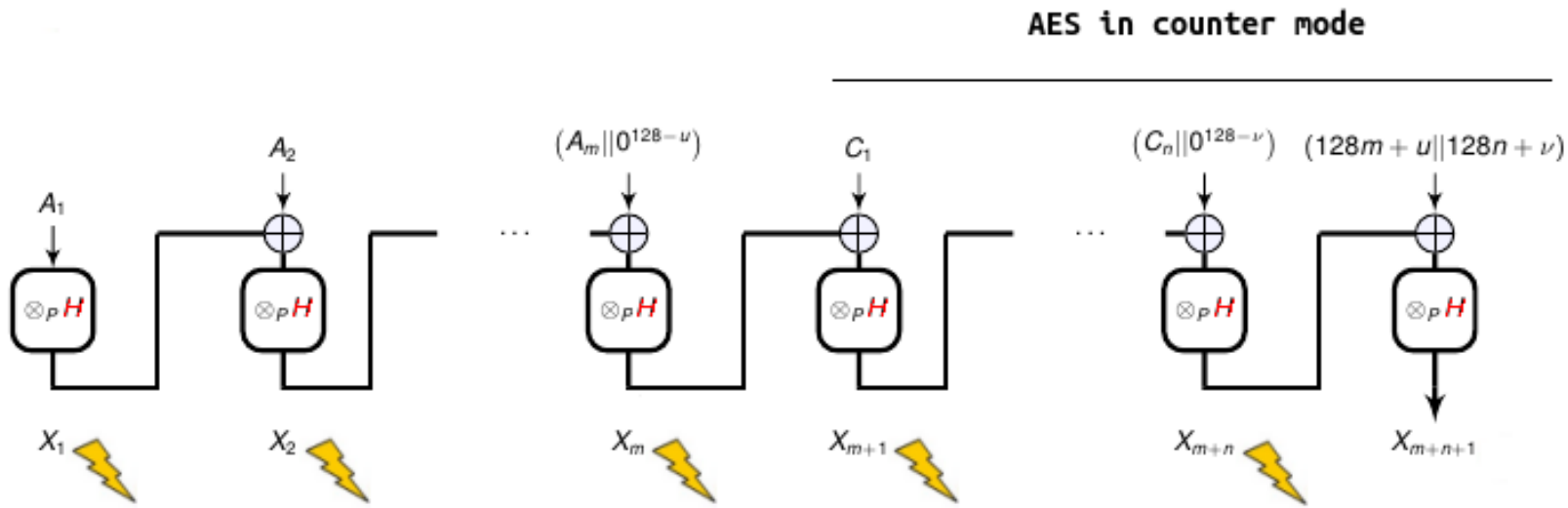
3. Saving more Executions



More Multiplications [Ferguson]:

$$
\begin{aligned}
X_1 &= M_1 \otimes_P H, \\
X_2 &= M_1 \otimes_P H^2 \oplus M_2 \otimes_P H, \\
X_3 &= M_1 \otimes_P H^3 \oplus M_2 \otimes_P H^2 \oplus M_3 \otimes_P H, \\
X_4 &= M_1 \otimes_P H^4 \oplus M_2 \otimes_P H^3 \oplus M_3 \otimes_P H^2 \oplus M_4 \otimes_P H.
\end{aligned}
$$

3. Saving more Executions

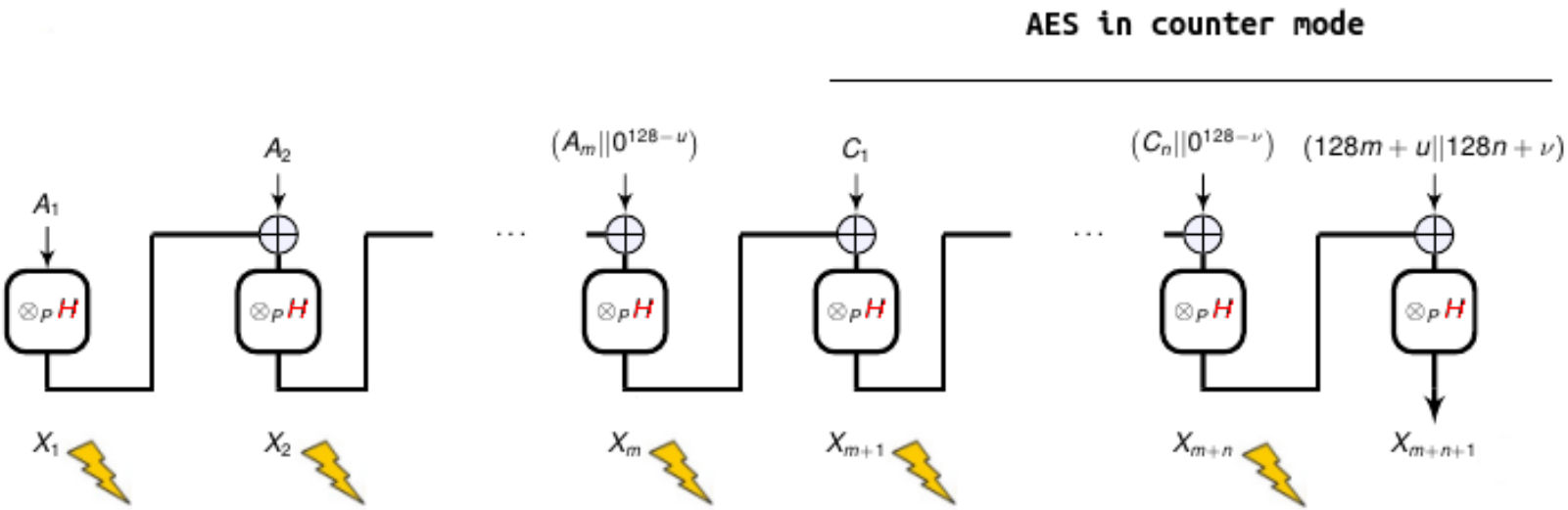


More Multiplications [Ferguson]:

$$
\begin{aligned}
X_1 &= M_1 \otimes_P H, \\
X_2 &= M_1 \otimes_P H^2 \oplus M_2 \otimes_P H, \\
X_3 &= M_1 \otimes_P H^3 \oplus M_2 \otimes_P H^2 \oplus M_3 \otimes_P H, \\
X_4 &= M_1 \otimes_P H^4 \oplus M_2 \otimes_P H^3 \oplus M_3 \otimes_P H^2 \oplus M_4 \otimes_P H.
\end{aligned}
$$

$$M_2 = 0$$

3. Saving more Executions



More Multiplications [Ferguson]:

- $X_1 = M_1 \otimes_P H,$
- $X_2 = M_1 \otimes_P H^2,$

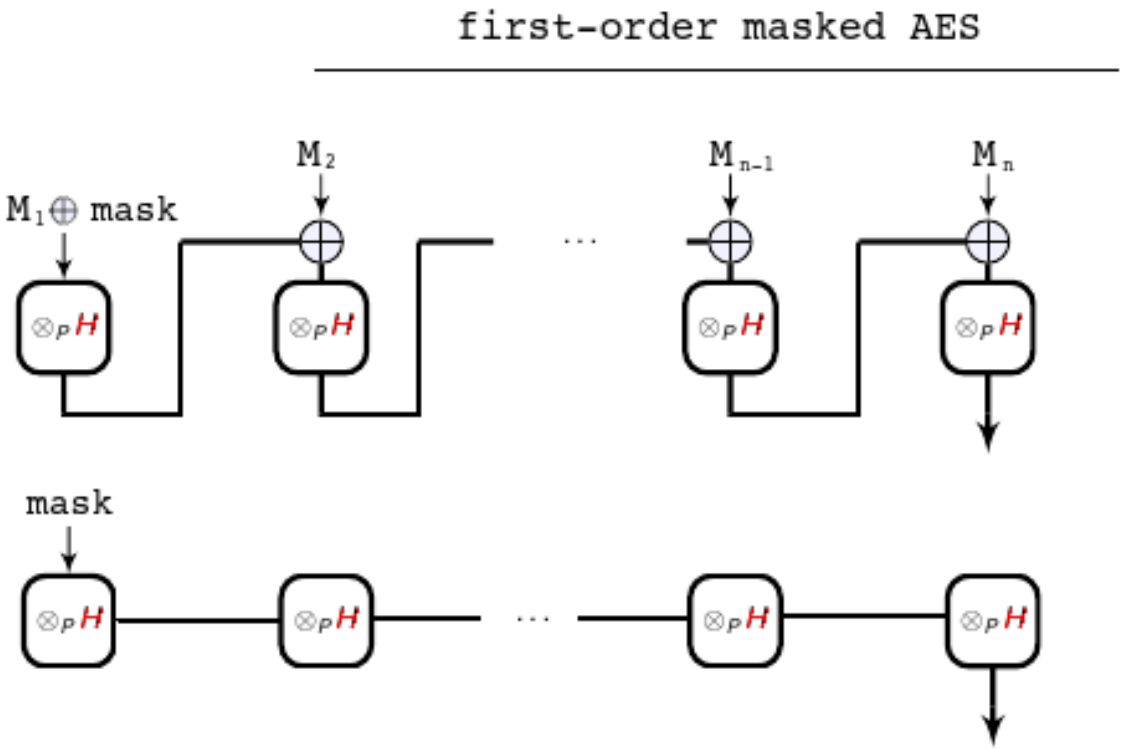
$X_3 = M_1 \otimes_P H^3 \oplus M_3 \otimes_P H,$

- $X_4 = M_1 \otimes_P H^4 \oplus M_3 \otimes_P H^2 \oplus M_4 \otimes_P H.$

$$M_2 = 0$$

Outline

Countermeasure

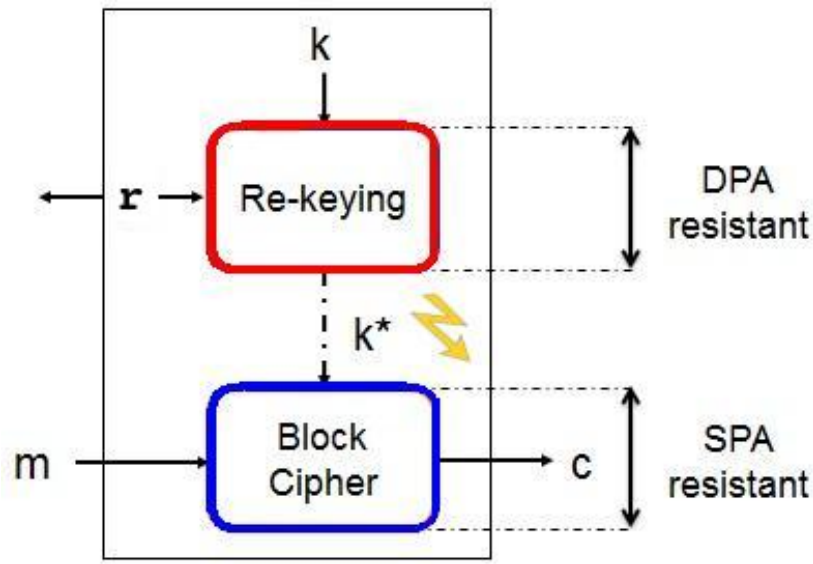


Complexity

- twice slower than the initial complexity
- additional generation of a 128-bit random value: mask
- *first-order masked AES: around 2.7 slower than the original AES*

Outline

Multiplication-based Re-keying[3]: Principle



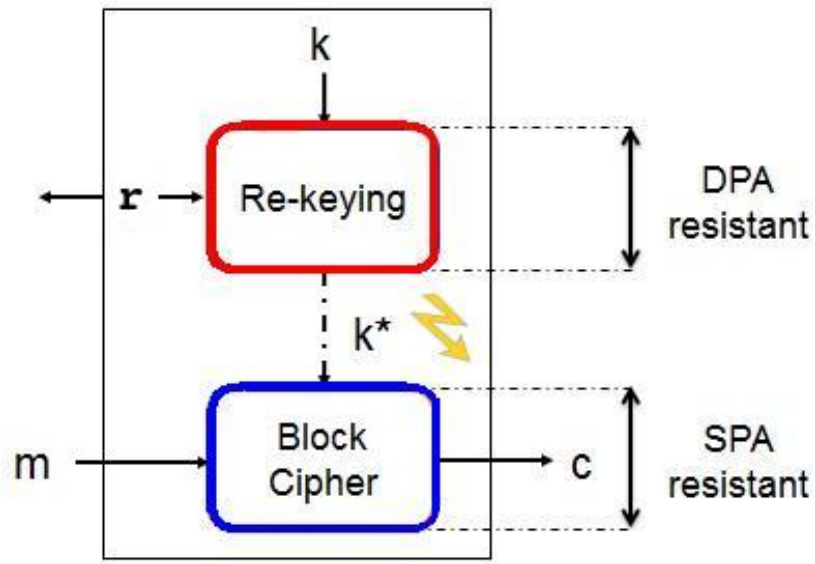
[3] M. Medwed, C. Petit, F. Regazzoni, M. Renauld, F.-X. Standaert, Fresh Re-Keying II: Securing Multiple Parties against Side-Channel and Fault Attacks, CARDIS 2011

Multiplication-based Re-keying[3]: Principle

Re-keying Primitive:

$$r \leftarrow \$; \quad k^\star \leftarrow r \otimes k$$
$$\text{in} \quad \mathrm{GF}(2^8)[y]/P(y) = y^{16} + 1$$

$$R_P \cdot k = \begin{pmatrix} r_0 k_0 & r_{15} k_1 & \cdots & r_1 k_{15} \\ r_1 k_0 & r_0 k_1 & \cdots & r_2 k_{15} \\ \vdots & \vdots & \ddots & \vdots \\ r_{15} k_0 & r_{14} k_1 & \cdots & r_0 k_{15} \end{pmatrix}$$



Block Cipher:

$$c \leftarrow BC_{k^\star}(m)$$

[3] M. Medwed, C. Petit, F. Regazzoni, M. Renauld, F.-X. Standaert, Fresh Re-Keying II: Securing Multiple Parties against Side-Channel and Fault Attacks, CARDIS 2011

Multiplication-based Re-keying: Attack ?

$$R_p \cdot k = \begin{pmatrix} r_0 k_0 & r_{15} k_1 & \cdots & r_1 k_{15} \\ r_1 k_0 & r_0 k_1 & \cdots & r_2 k_{15} \\ \vdots & \vdots & \ddots & \vdots \\ r_{15} k_0 & r_{14} k_1 & \cdots & r_0 k_{15} \end{pmatrix} = \begin{pmatrix} k_0^\star \\ k_1^\star \\ \vdots \\ k_{15}^\star \end{pmatrix}$$

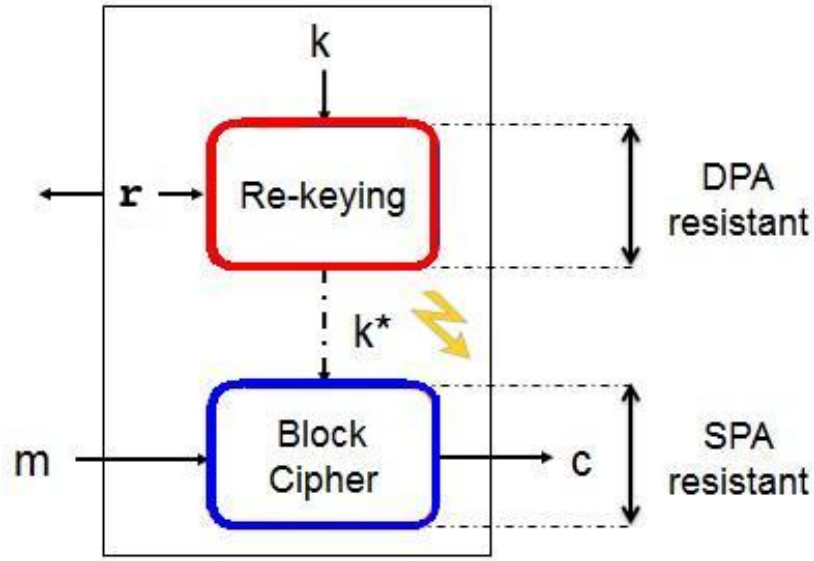
Equation of the LSB:

$$\mathrm{lsb}_0 \left(\mathrm{HW} \left[\left(\bigoplus_{0 \leqslant i \leqslant m-1} r_i \right) \cdot \left(\bigoplus_{0 \leqslant j \leqslant m-1} k_j \right) \right] \right) = b_0$$

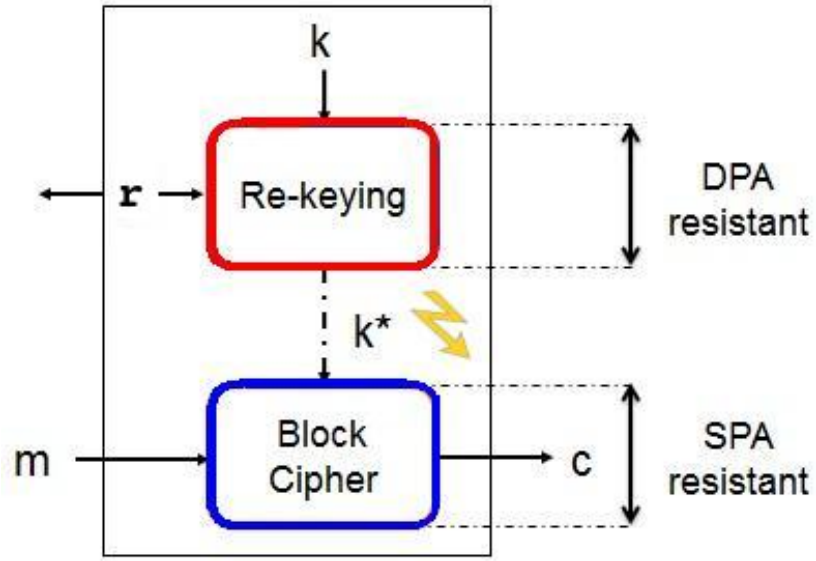
Recovery:

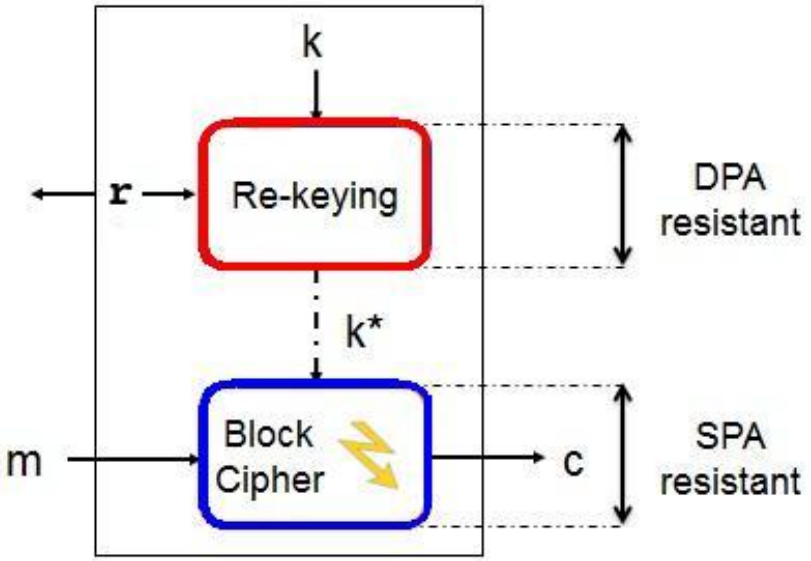
$$\rightarrow \quad \text{only} \left(\bigoplus_{0 \leqslant j \leqslant m-1} k_j \right)$$

Multiplication-based Re-keying: Better Attack

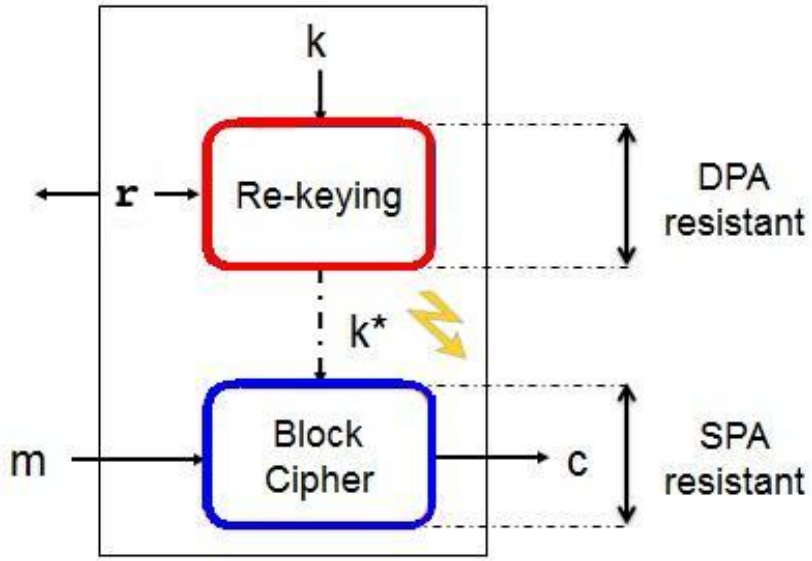


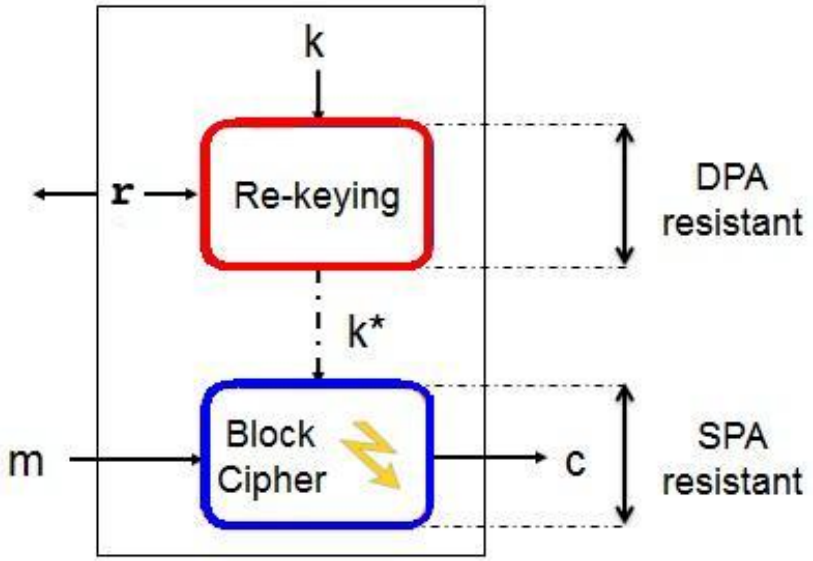
Multiplication-based Re-keying: Better Attack





Multiplication-based Re-keying: Better Attack





We have:

$$\mathrm{HW}(k_0^\star \oplus m_0)$$

with

$$(R_p \cdot k)_0 = \begin{pmatrix} r_0 k_0 & r_{15} k_1 & \cdots & r_1 k_{15} \end{pmatrix} = \begin{pmatrix} k_0^\star \end{pmatrix}$$

thus

$$\mathrm{lsb}_0 \left(\mathrm{HW} \left(\bigoplus_{0 \leqslant i \leqslant m-1} r_i k_i \oplus m_0 \right) \right) = b_0$$

Outline

Conclusion

- Summary
  - ⋆ attack the AES-GCM authentication without observing inside the multiplication
  - ⋆ different improvements
  - ⋆ adaptation of the attack on the multiplication-based re-keying
- Further Work
  - ⋆ application of similar attacks to other primitives
  - ⋆ exploitation of more leakage bits with different techniques

Thank you

Thank you for your attention.