

Improved Side-Channel Analysis of Finite-Field Multiplication

Sonia Belaïd¹ Jean-Sébastien Coron² Pierre-Alain Fouque³
Benoît Gérard⁴ Jean-Gabriel Kammerer⁵ Emmanuel Prouff⁶

¹École normale supérieure and Thales Communications & Security,

²University of Luxembourg

³Université de Rennes 1 and Institut Universitaire de France

⁴DGA.MI and IRISA

⁵DGA.MI and IRMAR

⁶ANSSI



Outline

Introduction

- Side-Channel Attacks
- Classical Power-Analysis Attacks
- Hidden Multiplier Problem
- State of The Art

New Attack

- Main Idea
- Filtering
- Solving the System with Errors
- Extension to Chosen Inputs

Conclusion

Outline

Introduction

- Side-Channel Attacks
- Classical Power-Analysis Attacks
- Hidden Multiplier Problem
- State of The Art

New Attack

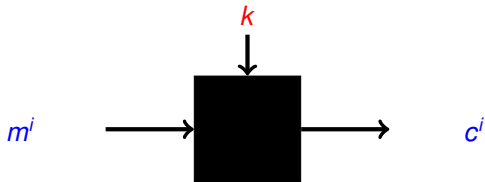
- Main Idea
- Filtering
- Solving the System with Errors
- Extension to Chosen Inputs

Conclusion

- Black-box cryptanalysis
- Side-channel analysis

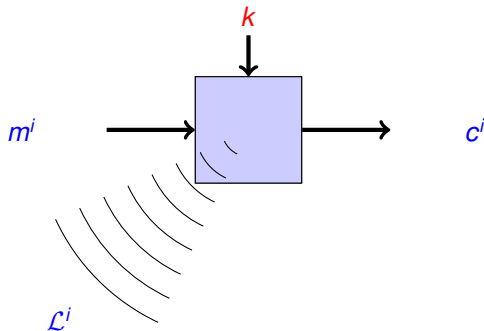
→ Black-box cryptanalysis: $\mathcal{A} \leftarrow (m^i, c^i)$

→ Side-Channel Analysis



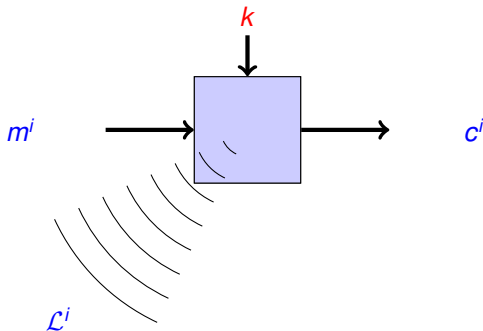
→ Black-box cryptanalysis

→ Side-Channel Analysis: $\mathcal{A} \leftarrow (m^i, c^i, \mathcal{L}^i)$



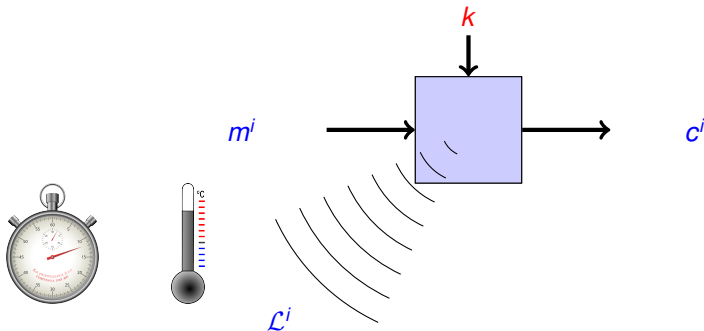
→ Black-box cryptanalysis

→ Side-Channel Analysis: $\mathcal{A} \leftarrow (m^i, c^i, \mathcal{L}^i)$



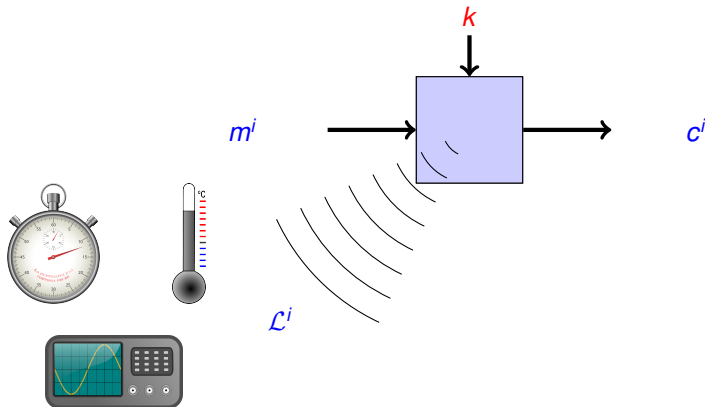
→ Black-box cryptanalysis

→ Side-Channel Analysis: $\mathcal{A} \leftarrow (m^i, c^i, \mathcal{L}^i)$



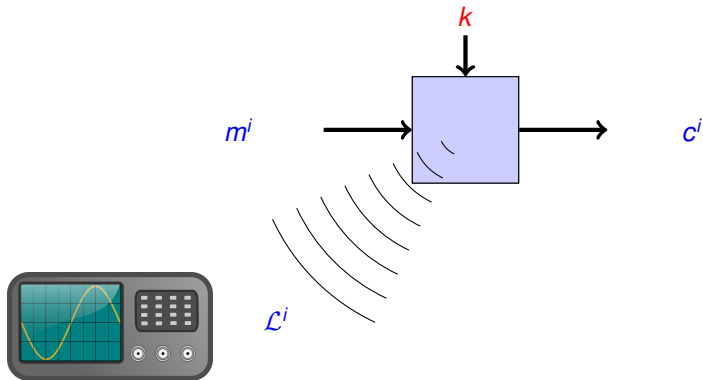
→ Black-box cryptanalysis

→ Side-Channel Analysis: $\mathcal{A} \leftarrow (m^i, c^i, \mathcal{L}^i)$



→ Black-box cryptanalysis

→ Side-Channel Analysis: $\mathcal{A} \leftarrow (m^i, c^i, \mathcal{L}^i)$



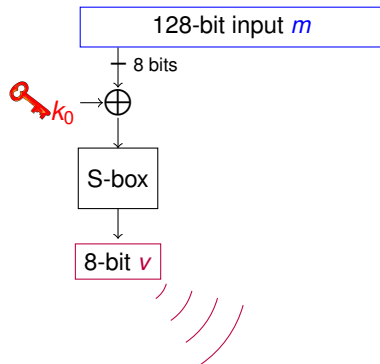
Classical Power-Analysis Attack against AES

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

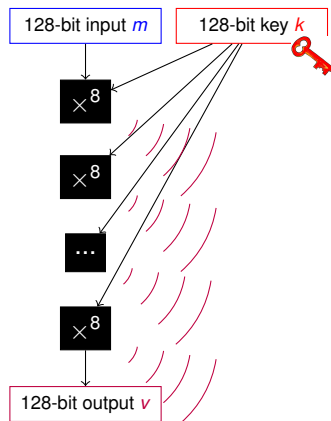
Attack on 128 bits

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



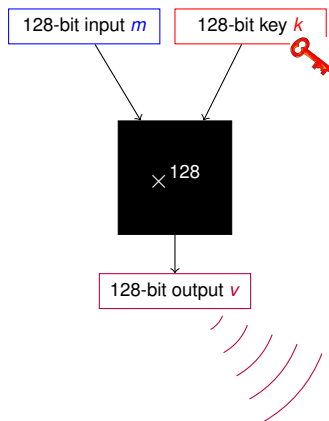
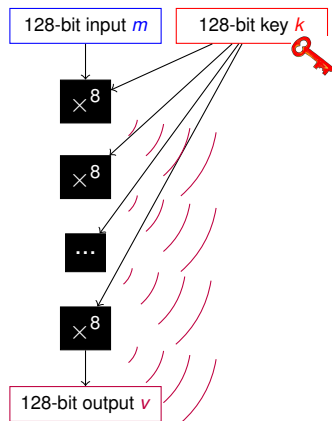
Power-Analysis Attack against AES-GCM authentication, multiplication-based fresh re-keying, ...

→ k is only manipulated in multiplications



Power-Analysis Attack against AES-GCM authentication, multiplication-based fresh re-keying, ...

→ k is only manipulated in multiplications



Hidden Multiplier Problem

Definition

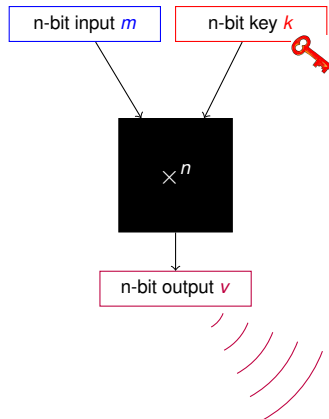
Let $k \leftarrow \text{GF}(2^n)$. Let $\ell \in \mathbb{N}$.

Given a sequence $\{m^i, \mathcal{L}^i\}_{1 \leq i \leq \ell}$ where

▶ $m^i \leftarrow \text{GF}(2^n)$

▶ $\mathcal{L}^i = \text{HW}(v^i) + \varepsilon^i, \varepsilon^i \sim \mathcal{N}(0, \sigma^2)$

recover k .



State of The Art



Sonia Belaïd, Pierre-Alain Fouque, and Benoît Gérard.

Side-channel analysis of multiplications in $\text{GF}(2^{128})$ - application to AES-GCM.

In Asiacrypt 2014, Proceedings, Part II, pages 306–325.

- use Hamming Weights' LSB
- solve a system with errors

Method	Signal-to-Noise Ratio = $\frac{\text{signal variance}}{\text{noise variance}} = 32/\sigma^2$			
	3.200	800	200	128
Naive method (C_s, C_t)	$(2^8, 2^{21})$	$(2^8, 2^{21})$	$(2^8, 2^{65})$	$(2^8, 2^{107})$
LPN (LF Algo) (C_s, C_t)	$(2^{11}, 2^{14})$	$(2^{20}, 2^{22})$	$(2^{32}, 2^{34})$	$(2^{48}, 2^{50})$
Linear decoding (C_s, C_t)	$(2^6, 2^6)$	$(2^6, 2^7)$	$(2^8, 2^{25})$	$(2^9, 2^{62})$

State of The Art



Sonia Belaïd, Pierre-Alain Fouque, and Benoît Gérard.

Side-channel analysis of multiplications in $\text{GF}(2^{128})$ - application to AES-GCM.

In Asiacrypt 2014, Proceedings, Part II, pages 306–325.

- use Hamming Weights' LSB
- solve a system with errors

Method	Signal-to-Noise Ratio = $\frac{\text{signal variance}}{\text{noise variance}} = 32/\sigma^2$			
	3.200	800	200	128
Naive method (C_s, C_t)	$(2^8, 2^{21})$	$(2^8, 2^{21})$	$(2^8, 2^{65})$	$(2^8, 2^{107})$
LPN (LF Algo) (C_s, C_t)	$(2^{11}, 2^{14})$	$(2^{20}, 2^{22})$	$(2^{32}, 2^{34})$	$(2^{48}, 2^{50})$
Linear decoding (C_s, C_t)	$(2^6, 2^6)$	$(2^6, 2^7)$	$(2^8, 2^{25})$	$(2^9, 2^{62})$

- ✗ specific to multiplication in $\text{GF}(2^{128})$
- ✗ highly impacted by noise

Outline

Introduction

- Side-Channel Attacks
- Classical Power-Analysis Attacks
- Hidden Multiplier Problem
- State of The Art

New Attack

- Main Idea
- Filtering
- Solving the System with Errors
- Extension to Chosen Inputs

Conclusion

Contributions

New Attack:

- filter the multiplication's outputs leakage to extract high and low Hamming weights
- solve a system with errors

Contributions

New Attack:

- filter the multiplication's outputs leakage to extract high and low Hamming weights
- solve a system with errors
- ✓ less impacted by noise
- ✓ more generic

Main Idea of The Attack

Reminder:

$$\mathcal{L}(v) = \text{HW}(v) + \varepsilon = \text{HW}(m \cdot k) + \varepsilon$$

Extreme cases:

$$\text{HW}(v) = 0 \rightarrow v = 0$$

$$\text{HW}(v) = n \rightarrow v = 2^n - 1$$

$$\left\{ \begin{array}{lcl} v_0 & = & \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(0,j)} m_i \right) k_j = 0 \\ v_1 & = & \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(1,j)} m_i \right) k_j = 0 \\ \vdots & & \vdots \\ v_{n-1} & = & \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(n-1,j)} m_i \right) k_j = 0 \end{array} \right.$$

$$\left\{ \begin{array}{lcl} v_0 & = & \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(0,j)} m_i \right) k_j = 1 \\ v_1 & = & \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(1,j)} m_i \right) k_j = 1 \\ \vdots & & \vdots \\ v_{n-1} & = & \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(n-1,j)} m_i \right) k_j = 1 \end{array} \right.$$

Main Idea of The Attack

Reminder:

$$\mathcal{L}(v) = \text{HW}(v) + \varepsilon = \text{HW}(m \cdot k) + \varepsilon$$

Usual cases:

$$\mathcal{L}(v) \text{ low} \rightarrow v \approx 0$$

$$\mathcal{L}(v) \text{ high} \rightarrow v \approx 2^n - 1$$

$$\left\{ \begin{array}{l} v_0 = \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(0,j)} m_i \right) k_j = 0 \\ v_1 = \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(1,j)} m_i \right) k_j = 0 \\ \vdots \quad \quad \quad \vdots \\ v_{n-1} = \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(n-1,j)} m_i \right) k_j = 0 \end{array} \right.$$

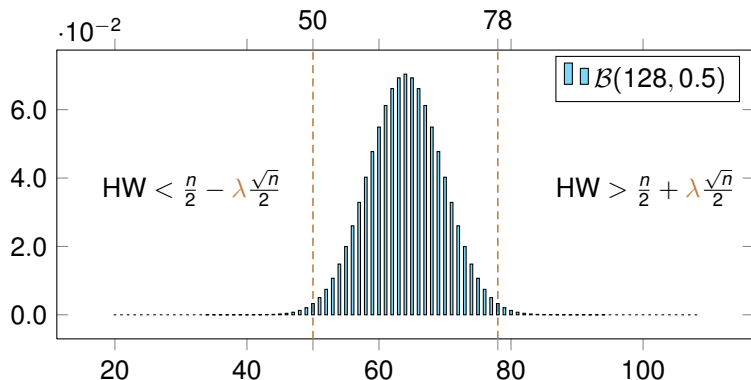
$$\left\{ \begin{array}{l} v_0 = \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(0,j)} m_i \right) k_j = 1 \\ v_1 = \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(1,j)} m_i \right) k_j = 1 \\ \vdots \quad \quad \quad \vdots \\ v_{n-1} = \bigoplus_{0 \leq j < n} \left(\bigoplus_{i \in I(n-1,j)} m_i \right) k_j = 1 \end{array} \right.$$

with an error probability p

Two Steps

1. filter the lowest and highest Hamming weights with a limited number of consumption traces to limit the error probability p
 - obtain a linear system with errors
2. solve the system with the error probability p
 - recover the secret key k

Step 1: Filtering



$$SNR = 128$$

$$n = 128$$

$$\lambda \approx 2.5$$

filtering: 1 trace over 2^5
error probability: $p \approx 0.38$

Step 1: Filtering

Proportion of filtered acquisitions:

$$F(\lambda, \sigma) = 1 - 2^{-n} \sum_{y=0}^n \binom{n}{y} \int_{n/2-\lambda s}^{n/2+\lambda s} \phi_{y,\sigma}(t) dt, \quad \text{with } s = \sqrt{n}/2$$

Error probability:

$$p(\lambda, \sigma) = \frac{1}{F(\lambda, \sigma)} \sum_{y=0}^n \frac{\binom{n}{y}}{2^n} \left(\underbrace{\frac{y}{n} \int_{-\infty}^{n/2-\lambda s} \phi_{y,\sigma}(t) dt}_{\text{low Hamming weights}} + \underbrace{\left(1 - \frac{y}{n}\right) \int_{n/2+\lambda s}^{+\infty} \phi_{y,\sigma}(t) dt}_{\text{high Hamming weights}} \right)$$

Step 1: Filtering

$\log_2(1/F(\lambda))$	30	25	20	15	10	5
SNR = 128, $\sigma = 0.5$						
λ	6.00	5.46	4.85	4.15	3.29	2.16
p	0.23	0.25	0.28	0.31	0.34	0.39
p [BFG14]	0.31					
SNR = 8, $\sigma = 2$						
λ	6.37	5.79	5.14	4.39	3.48	2.28
p	0.25	0.27	0.29	0.32	0.35	0.40
p [BFG14]	> 0.49					
SNR = 2, $\sigma = 4$						
λ	7.42	6.73	5.97	5.09	4.03	2.64
p	0.28	0.30	0.32	0.34	0.37	0.41
p [BFG14]	> 0.49					
SNR = 0.5, $\sigma = 8$						
λ	10.57	9.58	8.48	7.21	5.71	3.73
p	0.34	0.36	0.37	0.39	0.41	0.44
p [BFG14]	> 0.49					

Step 2: Solving the System with Errors

Classical LPN problem: recover the secret key from a noisy system

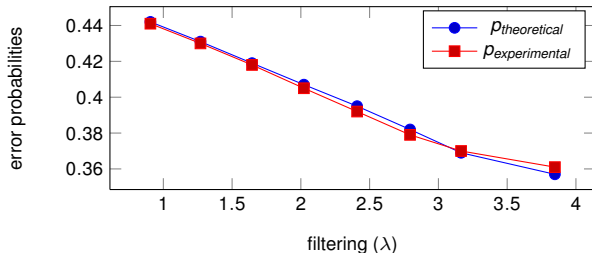
- limited memory
- limited computational power

Specific constraints:

- limited number of equations/consumption traces
- key size n (e.g., 128)
- probability of errors dependent on the filtering and on the noise

Experiments

- ▶ Filtering on a Virtex 5 (128 bits) : $\text{SNR} = 8.21$, $\sigma = 7.11$



- ▶ Expected complexities to recover k with 2^{20} traces ($p \approx 0.29$)

	$(2^{59.31}, 2^{27.00})$
(time , memory)	$(2^{51.68}, 2^{36.00})$
	$(2^{50.00}, 2^{44.00})$

Extension: Chosen Inputs in $\text{GF}(2^{128})$

1. Exhibit the noisy system:

► $\text{MSB}(m \cdot k) = 0 \rightarrow \text{HW}((2 \cdot m) \cdot k) = \text{HW}(m \cdot k)$

► $\text{MSB}(m \cdot k) = 1 \rightarrow$

$$|\text{HW}((2 \cdot m) \cdot k) - \text{HW}(m \cdot k)| = \begin{cases} 1 & \text{with probability} = 3/4 \\ 3 & \text{with probability} = 1/4 \end{cases}$$

SNR (σ)	128 (0.5)	8 (2)	2 (4)	0.5 (8)
p	0.003	0.27	0.39	0.46

Extension: Chosen Inputs in $GF(2^{128})$

1. Exhibit the noisy system:

► $MSB(m \cdot k) = 0 \rightarrow HW((2 \cdot m) \cdot k) = HW(m \cdot k)$

► $MSB(m \cdot k) = 1 \rightarrow$

$$|HW((2 \cdot m) \cdot k) - HW(m \cdot k)| = \begin{cases} 1 & \text{with probability} = 3/4 \\ 3 & \text{with probability} = 1/4 \end{cases}$$

SNR (σ)	128 (0.5)	8 (2)	2 (4)	0.5 (8)
p	0.003	0.27	0.39	0.46

2. Solve the noisy system:

- only 128 equations
- repetitions to obtain a system with almost no error

Example:

- SNR of 128 can be achieved from an SNR of 2 and 64 repetitions
- $128 \times 0.003 = 0.384$ errors
- solving the system with a single error: 2^7 key verifications

Outline

Introduction

- Side-Channel Attacks
- Classical Power-Analysis Attacks
- Hidden Multiplier Problem
- State of The Art

New Attack

- Main Idea
- Filtering
- Solving the System with Errors
- Extension to Chosen Inputs

Conclusion

Conclusion

Summary

- ★ attack on multiplications without looking inside the multiplication
- ★ less noise sensitive than [BFG14]
- ★ practical for $n = 128$

Further Work

- ★ application of similar attacks to other primitives
- ★ deeper analysis of LPN techniques to improve solving in side-channel contexts

Conclusion

Summary

- ★ attack on multiplications without looking inside the multiplication
- ★ less noise sensitive than [BFG14]
- ★ practical for $n = 128$

Further Work

- ★ application of similar attacks to other primitives
- ★ deeper analysis of LPN techniques to improve solving in side-channel contexts

Thank you for your attention.